

Open Research Online

The Open University's repository of research publications and other research outputs

Operational support systems for satellite communications

Thesis

How to cite:

Fitch, Michael Robert (2005). Operational support systems for satellite communications. PhD thesis The Open University.

For guidance on citations see [FAQs](#).

© 2005 Michael Robert Fitch



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Version: Accepted Manuscript

Link(s) to article on publisher's website:

<http://dx.doi.org/doi:10.21954/ou.ro.000101b7>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

Michael Robert Fitch BA(Hons)

**Operational Support Systems for Satellite
Communications**

A thesis submitted for the degree of
Doctor of Philosophy

**Department of Information and Communication
Technologies
of the
Open University**

2005

ProQuest Number: 13917259

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 13917259

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Abstract

The role of satellite communications is changing from providing bandwidth linking network operators interconnections towards providing IP enabled communications to end users. This migration from few high-value routes towards many low-value routes means that integration and automation of processes with terrestrial networks becomes critical in driving down unit costs. Integration and automation is necessary on all planes: user, control and management. In satellite communications, management aspects, underpinned by Operational Support Systems (OSS) have received the least research attention, making this a valuable topic for study.

In most areas, OSS for satellite systems are similar to other domains. However there are some notable areas of difference which have been the focus of this research. The eTOM business framework, developed by the TMF, has been used to highlight aspects of OSS unique to satellite. Since satellite capacity represents the highest operational cost of a satellite route, effective management while minimising the overhead traffic is critical. The transmission of IP packets is assumed and the real-time measurement of QoS parameters such as packet delay and loss emerged as the most important differences.

A number of approaches to QoS measurement are feasible, however the use of trace packets is most promising especially for high network loads. An experiment compares the results from simulations, mathematical models and from a test network, using Poisson and self-similar traffic flows. The relationship between measurement accuracy and trace packet intensity is explored and the measurement response time to steps in traffic load is estimated. It is discovered that measurement accuracy improves as the queue load increases, in contrast to alternative approaches such as sampling of user packets. The response time to steps depends upon the degree of self-similarity and is generally longer than the times recommended by standards. A pragmatic approach to management of different modes is proposed where the measurement method is changed depending on the load.

Table of contents

Abstract	ii
Table of contents	iii
1 Introduction	1-1
1.1 Background	1-1
1.2 The need for this research	1-3
1.3 The research questions	1-3
1.4 Structure of the thesis	1-4
1.5 References for chapter 1	1-7
2 Overview of satellite systems	2-1
2.1 Introduction	2-1
2.2 What satellites are	2-1
2.3 Satellite link characteristics	2-2
2.3.1 Wide area coverage	2-2
2.3.2 Typical transmission rates	2-2
2.3.3 Propagation delay	2-2
2.3.4 Outages due to bad weather	2-3
2.3.5 Limited power and bandwidth	2-4
2.3.6 High cost and limited life	2-4
2.4 Technical overview of satellite systems	2-4
2.4.1 Frequency bands	2-4
2.4.2 Transparent satellites	2-5
2.4.3 Processing satellites	2-7
2.4.4 Orbit type	2-9
2.4.5 Number of satellites required	2-11
2.4.6 Telemetry, Telecommand and Control	2-11
2.4.7 Multiple access by user terminals	2-12
2.4.8 Demand-assigned multiple access (DAMA)	2-13
2.4.9 Reliable broadcast and multicast	2-14
2.4.10 Doppler and network synchronisation	2-15
2.5 The changing use of satellite	2-16
2.5.1 Changes in the marketplace	2-16
2.5.2 Satellite operator response to the changes in the marketplace	2-17
2.6 Some example systems	2-19
2.7 References for chapter 2	2-21
3 Overview of Operational Support Systems (OSS) and their relationship with satellite services	3-1
3.1 Introduction	3-1

3.2	What is OSS	3-1
3.3	OSS key business drivers	3-3
3.4	Review of OSS architectures and technology	3-5
3.4.1	The management pyramid.....	3-5
3.4.2	OSS protocols and the role of managers and agents.....	3-6
3.4.3	Distributed architectures and associated software	3-9
3.4.4	Web services	3-11
3.5	Examples of COTS software.....	3-12
3.6	Review of OSS industry models	3-13
3.6.1	ITU-T	3-13
3.6.2	Parlay	3-15
3.6.3	TINA	3-15
3.6.4	The Telemanagement Forum (TMF)	3-16
3.7	An overview of relevant OSS research	3-19
3.8	The value chain	3-22
3.8.1	Derivation of basic roles and actors	3-22
3.8.2	The service level agreement.....	3-23
3.8.3	OSS and signalling flows.....	3-24
3.8.4	Management boundaries	3-25
3.9	Analysis of differences between satellite and terrestrial OSS processes	3-29
3.9.1	The eTOM level 1 processes.....	3-29
3.9.2	Level 2 processes	3-31
3.9.3	Identification and prioritisation of processes relevant to satellite.....	3-34
3.9.4	Mapping of processes onto satellite architectures.....	3-36
3.9.5	Identification of areas for further study	3-39
3.9.6	Conclusion of evaluation	3-39
3.10	References for chapter 3.....	3-41
4	Management and measurement of QoS parameters.....	4-1
4.1	Introduction.....	4-1
4.2	QoS management process and management modes	4-1
4.3	Choice of QoS parameters relevant to satellite.....	4-2
4.3.1	Assumptions made and rationale	4-2
4.3.2	QoS parameter selection	4-5
4.3.3	The measurement of delay, delay variation and packet loss.....	4-7
4.4	State-of-the-art on QoS parameter measurements and management	4-9
4.4.1	Research activity on QoS management.....	4-10
4.4.2	Survey of tools for measuring packet delay and loss.....	4-15
4.4.3	Use of trace packets	4-17
4.5	Summary of chapter 4	4-17
4.6	References for chapter 4.....	4-19
5	Description of models and test network.....	5-1
5.1	Introduction.....	5-1
5.2	Overview of OPNET.....	5-1
5.3	Traffic models.....	5-4
5.3.1	A Poisson model	5-4

5.3.2	A self-similar model.....	5-5
5.4	Queueing model	5-12
5.4.1	Description	5-12
5.4.2	Calibration of the OPNET model.....	5-16
5.4.3	Waiting time distribution	5-18
5.5	The test network.....	5-21
5.5.1	Description	5-21
5.5.2	Network set-up and test.....	5-26
5.6	Summary of chapter 5	5-30
5.7	References for chapter 5.....	5-31
6	Experimental results: the use of trace packets to measure user packet delay and loss.....	6-1
6.1	Introduction	6-1
6.2	Measurement parameters	6-2
6.2.1	User packet generation patterns	6-2
6.2.2	Trace packet generation	6-3
6.2.3	Satellite link rate and delay	6-3
6.2.4	Measurement times	6-3
6.2.5	Queue type and length.....	6-4
6.3	Effect of limiting the queue length	6-4
6.3.1	The effect of a queue limit on Poisson traffic	6-4
6.3.2	The effect of a queue limit on self-similar traffic	6-5
6.4	Simulation results of estimation accuracy, Poisson source.....	6-9
6.4.1	Effect on delay caused by adding trace packets.....	6-9
6.4.2	Absolute measurements of error	6-10
6.4.3	Plotting of results against trace packet variance	6-12
6.4.4	Different length user packets: 64 and 1500 bytes	6-14
6.4.5	Estimation of packet loss	6-16
6.5	Simulation results of estimation accuracy, self-similar source.....	6-17
6.5.1	Estimation of delay	6-17
6.5.2	Estimation of packet loss	6-18
6.6	Test network results, Poisson and self-similar sources.....	6-18
6.6.1	Estimation of packet delay, Poisson traffic.....	6-18
6.6.2	Estimation of packet loss, Poisson traffic	6-20
6.6.3	Estimation of packet delay, self-similar traffic	6-20
6.6.4	Estimation of packet loss, self-similar traffic	6-21
6.7	Investigation into faster estimation times	6-22
6.7.1	Response of the queue to steps in load.....	6-22
6.7.2	Simulation results for detection of steps	6-23
6.7.3	Test network results for detection of steps.....	6-26
6.7.4	Discussion on detection of step changes in load.....	6-29
6.8	Use of trace packets compared to other methods: direct sampling and SNMP	6-29
6.8.1	Comparison with direct sampling	6-29
	Confidence in the estimation of delay mean	6-30
	Confidence in the estimation of variance.....	6-31
6.8.2	Comparison with SNMP commands.....	6-33
6.9	Summary of the chapter	6-33

6.9.1	Measurement modes	6-33
6.9.2	Comparison of behaviour of self-similar and Poisson traffic	6-34
6.9.3	The use of trace packets as estimators of packet delay and loss	6-34
6.10	References for chapter 6.....	6-36
7	Discussion and conclusions	7-1
7.1	The changing role of satellite links and implications on OSS	7-1
7.2	Evaluation of differences between OSS for terrestrial and satellite networks and services.....	7-3
7.3	Measurement and management of QoS parameters.....	7-4
7.4	Simulation model and test network.....	7-5
7.5	The use of trace packets as estimators	7-6
7.6	Management modes	7-8
7.7	Faster measurement times	7-9
7.8	Further work.....	7-10
7.9	Concluding remarks	7-11
Appendix A	OPNET models and code	A-1
A.1	Overall model.....	A-1
A.2	Traffic servers	A-1
A.2.1	Poisson and trace packet source	A-2
A.2.2	Self-similar source	A-5
A.3	Gateway model.....	A-10
A.4	Packet switch model.....	A-14
Appendix B:	Basic queuing theory	B-1
B.1	Queueing theory	B-1
B.2	Derivation of the Poisson arrival process.....	B-3
B.3	Derivation of time spent in the system.....	B-4
B.4	References	B-7
Appendix C	Test network router configurations and RUDE files	C-1
C.1	Test network.....	C-1
C.2	Router Configurations.....	C-1
C.3	RUDE Files	C-2
Table of figures	vii
Table of tables	x
Glossary	xi
Publications related to this research	xvi
Acknowledgements	xvii

Table of figures

Figure 1-1 Structure of the study	1-5
Figure 2-1 A simple generic transparent payload	2-6
Figure 2-2 A simple generic processing payload	2-7
Figure 2-3 A typical arrangement for clocking and Doppler buffering on a satellite link.....	2-15
Figure 2-4 Where satellites are used	2-16
Figure 2-5 Money flow when a co-operative launches a satellite.....	2-18
Figure 3-1 The OSS S-curve	3-4
Figure 3-2 Management pyramid.....	3-5
Figure 3-3 Cascaded element managers.....	3-5
Figure 3-4 Location of MIB and agent	3-7
Figure 3-5 Use of a proxy agent.....	3-7
Figure 3-6 Role of IDL Compiler (shown with CORBA)	3-10
Figure 3-7 An integrated OSS structure.....	3-12
Figure 3-8 Role of Parlay API	3-15
Figure 3-9 The TOM.....	3-17
Figure 3-10 The eTOM level 1 processes framework	3-18
Figure 3-11 A fundamental diagram of information flow and payment.....	3-23
Figure 3-12 Making use of a Service Provider	3-23
Figure 3-13 The use of an Integrating Service Provider to provide a service mix	3-24
Figure 3-14 Showing OSS and Signalling paths.....	3-25
Figure 3-15 A practical example.....	3-26
Figure 3-16 Tunnelling OSS traffic over a satellite link.....	3-27
Figure 3-17 Mapping of FSAN, ATM Forum, DAVIC and UMTS Network Interface Nodes	3-28
Figure 3-18 eTOM level 1 processes	3-30
Figure 3-19 where satellites are used	3-37
Figure 4-1 Management process	4-1
Figure 4-2 Remote users connected by 2-way satellite link	4-3
Figure 4-3 Bit-error ratio curves (y-axis) against E_b/N_0	4-4
Figure 4-4 Communications path with two measurement points.....	4-7
Figure 4-5 Example of priority queuing	4-14

Figure 4-6 The trace packet method of measurement.....	4-17
Figure 5-1 OPNET network level design example	5-2
Figure 5-2 OPNET node level design example.....	5-2
Figure 5-3 OPNET process model design example	5-3
Figure 5-4 The Pareto distribution	5-6
Figure 5-5 Arrangement of ON-OFF traffic sources	5-6
Figure 5-6 Packets are produced at a constant rate during ON times from each source.....	5-7
Figure 5-7 Illustration of aggregation process	5-8
Figure 5-8a Block size $m = 100$	5-10
Figure 5-9 Plot of $\text{Log Var}(x)$ against $\text{Log } m$	5-12
Figure 5-10 Simple queue and server system	5-12
Figure 5-11 Calibration process using M/D/1 and M/M/1	5-13
Figure 5-12 Simple OPNET queue and server system	5-16
Figure 5-13 Example time-series delay for $\rho = 0.95$, M/D/1 system.....	5-17
Figure 5-14 Comparison of simulation results with theory – delay mean and variance for M/D/1 system.....	5-18
Figure 5-15 Comparison of simulation results with theory, system delay pdf for M/M/1	5-19
Figure 5-16 M/D/1 system delay pdfs from simulation histograms, with fitted curves	5-19
Figure 5-17 OPNET model for User plus Trace packets	5-20
Figure 5-18 Test network diagram.....	5-21
Figure 5-19 Photograph of test network equipment.....	5-22
Figure 5-20 Construction of IEEE802.3 Ethernet frame	5-23
Figure 5-21 Additional components for the live satellite checks.....	5-24
Figure 5-22 Photograph of 5.6m antenna.....	5-25
Figure 5-23 Photograph of rear of antenna showing Low Noise Amplifier (LNA)	5-25
Figure 5-24 Equipment rack inside earth station	5-26
Figure 5-25 Histogram of inter-arrival times of test packet burst.....	5-28
Figure 5-26 Histogram of inter-arrival times with no queuing	5-29
Figure 6-1 Layout of the chapter.....	6-1
Figure 6-2 Time series of delay, 100 Mbit queue	6-5
Figure 6-3 Time series of delay, 215 kbit queue.....	6-5
Figure 6-4 Time series of delay, 100 Mbit queue	6-6
Figure 6-5 Delay histogram, 100 Mbit queue	6-6
Figure 6-6 Time series of delay, user packets, 215 kbit queue	6-7
Figure 6-7 Time series of delay, trace packets, 215 kbit queue.....	6-7
Figure 6-8 Delay histogram, user packets, 215 kbit queue	6-7
Figure 6-9 Delay histogram, trace packets, 215 kbit queue	6-8
Figure 6-10 Effect on user packet delay mean and variance caused by adding trace packets .	6-9
Figure 6-11 Delay mean against number of trace packets used for estimation	6-10
Figure 6-12 Delay variance against number of trace packets used for estimation.....	6-11
Figure 6-13 Delay pdf of user and trace packets at load $\rho = 0.95$	6-12
Figure 6-14 Percentage error in estimation of user packet delay mean against trace packet variance	6-13
Figure 6-15 Percentage error in estimation of user packet delay variance against trace packet variance	6-13

Figure 6-16 Error in estimation of user packet delay mean, 64 byte user packets	6-14
Figure 6-17 Error in estimation of user packet delay variance, 64 byte user packets.....	6-14
Figure 6-18 Error in estimation of user packet delay mean, 1500 byte user packets	6-15
Figure 6-19 Error in estimation of user packet delay variance, 1500 byte user packets.....	6-15
Figure 6-20 Error in estimation of user packet delay mean.....	6-17
Figure 6-21 Error in estimation of user packet delay variance	6-17
Figure 6-22 Error in estimation of user packet delay mean.....	6-19
Figure 6-23 Error in estimation of user packet delay variance	6-19
Figure 6-24 Error in estimation of user packet delay mean.....	6-21
Figure 6-25 Error in estimation of user packet delay variance	6-21
Figure 6-26. Response in delay for step increase of ρ from 0.5 to 0.9. User packets from Poisson source.....	6-22
Figure 6-27. Response in delay for step increase of ρ from 0.3 to 0.7. User packets from self-similar source	6-23
Figure 6-28 10*10-second estimations of mean, Poisson traffic	6-24
Figure 6-29. 10*10-second estimations of variance, Poisson traffic	6-24
Figure 6-30 10*10-second estimations of mean, self-similar traffic	6-24
Figure 6-31 10*10-second estimations of variance, self-similar traffic	6-25
Figure 6-32 10*100-second estimations of mean, Poisson traffic	6-25
Figure 6-33 10*100-second estimations of variance, Poisson traffic	6-25
Figure 6-34 10*100-second estimations of mean, self-similar traffic	6-26
Figure 6-35 10*100-second estimations of variance, self-similar traffic	6-26
Figure 6-36 10 * 10 second estimations of delay mean, Poisson traffic.....	6-27
Figure 6-37 10 * 10 second estimations of delay variance, Poisson traffic	6-27
Figure 6-38 10 * 100 second estimations of delay mean, Poisson traffic.....	6-28
Figure 6-39 10 * 100 second estimations of delay variance, Poisson traffic	6-28
Figure 6-40 5 * 100 second estimations of delay mean, self-similar traffic	6-28
Figure 6-41 5 * 100 second estimations of delay variance, self-similar traffic	6-29
Figure 6-42 Chi-square distribution.....	6-32

Table of tables

Table 2-1 Common satellite frequency bands and example services	2-5
Table 2-2 Some satellite systems	2-20
Table 3-1 eTOM Layer 0, 1 and 2 processes that are different for satellite and terrestrial systems	3-36
Table 3-2 Mapping of rank = 1 eTOM Level 2 processes mapped onto satellite link types	3-38
Table 4-1 Preliminary management modes	4-2
Table 4-2 Some satellite services and required QoS parameters	4-6
Table 4-3 Extract from ITU recommendation Y.1541 table 1	4-9
Table 4-4 Some results of estimation accuracy against sample intensity	4-13
Table 5-1 Least squares fit for pdfs of M/D/1	5-20
Table 6-1 Packet lengths used in the experiments	6-2
Table 6-2 Relationship of packet length to packet throughput	6-3
Table 6-3 Effect of queue limit on delay and loss characteristics, $\rho = 0.95$, Poisson source ..	6-5
Table 6-4 Effect of queue limit on user packet delay and loss characteristics, $\rho = 0.7$, self-similar source	6-8
Table 6-5 The effect of trace packets on user packet delay for $\rho = 0.95$	6-10
Table 6-6 Summary of results of delay estimation accuracy with different user packet lengths, Poisson, simulation	6-16
Table 6-7 Packet loss estimation error, Poisson traffic, $\rho = 0.95$, simulation	6-16
Table 6-8 Packet loss estimation error, self-similar traffic, $\rho = 0.7$, simulation	6-18
Table 6-9 Packet loss estimation error, Poisson traffic, test network, $\rho = 0.95$	6-20
Table 6-10 Packet loss estimation error, self-similar traffic, test network, $\rho = 0.7$	6-21
Table 6-11 Figure comparison table for steps	6-27
Table 6-12 Comparison of estimation accuracy of delay mean with confidence intervals ...	6-31
Table 6-13 Comparison of estimation accuracy of delay variance with confidence intervals...	6-32
Table 6-14 Management mode guidelines	6-34

Glossary

AAL	ATM Adaptation Layer
ACK	Acknowledgement
ADSL	Asymmetric Digital Subscriber Line
AN	Access Node
ANI	ATM Network Interworking
API	Applications Programmable Interface
ARP	Address Resolution Protocol
ASN.1	Abstract Syntax Notation 1
ATM	Asynchronous Transfer Mode
B2B	Business-to-Business
BER	Bit Error Ratio
BFN	BeamForming Network
BGP	Border Gateway Protocol
B-ISDN	Broadband ISDN
CAIDA	Co-operative Association for Internet Data Analysis
CAPEX	Capital Expenditure
CDV	Cell Delay Variation
CDMA	Code Division Multiple Access
CIO	Chief Information Officer
CLR	Cell Loss Ratio
CMIP	Common Management Interface Protocol
CMIS	Common Management Information Services
CMISE	Common Management Information Service Element
COM	Component Object Model
COPS	Common Open Policy Server
CORBA	Common Object Request Broker Architecture
COST	CO-operation europeenne dans le domaine de la recherche Scientifique et Technique
COTS	Commercial off-the-shelf
CRM	Customer Relationship Management
CTD	Cell Transfer Delay
DAMA	Demand Assigned Multiple Access
DASS	Digital Access Signalling System
DAVIC	Digital Audio Visual Council
DCE	Data Communications Equipment
DCE	Distributed Computing Environment
DCOM	Distributed COM
DCT	Discrete Cosine Transform
DIFFSERV	Differentiated Services
DII	Dynamic Invocation Interface

DPNSS	Digital Private Network Signalling System
DSLAM	Digital Subscriber Line Access Multiplexer
DVB-S	Digital Video Broadcasting - Satellite
DVB RCS	DVB Return Channel by Satellite
EPSRC	Engineering and Physical Sciences Research Council
ESA	European Space Agency
eTOM	Enterprise TOM
ETSI	European Technical Standards Institute
FAB	Fulfilment, Assurance and Billing
FCAPS	Fault, Configuration, Accounting, Performance and Security
FCFS	First Come First Served
FDMA	Frequency Division Multiple Access
FEC	Forward Error-correcting Code
FIFO	First In First Out
FSAN	Full Services Access Network
FSS	Fixed Satellite Service
FTP	File Transfer Protocol
GDMO	Guidelines for the Design of Managed Objects
GEO	Geostationary Orbit
GPS	Global Positioning by Satellite
GSM	Global System for Mobile communications
GSO	GeoSynchronous Orbit
HTTP	HyperText Transfer Protocol
ICMP	Internet Control Message Protocol
ICO	Intermediate Circular Orbit (also the name of a company that operates satellites in this orbit)
IEEE	Institutions of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IDD	International Direct Dial
IDL	Interface Definition Language
IDR	Intermediate Data Rate
IIOP	Internet Inter-ORB Protocol
IMS	IP Multimedia Subsystem
INTSERV	Integrated Services
IP	Internet Protocol
IPDV	Internet Packet Delay Variation
IPPM	IP Performance Metric (an IETF working group)
IPTD	IP Transfer Delay
ISDN	International Switched Data Network
iSP	Integrating Service Provider
ISP	Internet Service Provider
ITSO	International Telecommunications Satellite Organisation

ITU-T	International Telecommunications Union, Telecommunications Sector
JIDM	Joint InterDomain working group
LAN	Local Area Network
LEO	Low Earth Orbit
LNA	Low Noise Amplifier
MEO	Medium Earth Orbit
MF-TDMA	Multiple Frequency Time Division Multiple Access
MIB	Management Information Base
MP	Measurement Point
MSS	Mobile Satellite Service
NAK	Negative Acknowledgement
NGN	Next Generation Networks
NGOSS	New Generation OSS
NMS	Network Management System
NNI	Network to Network Interface
NOCC	Network Operations Control Centre
NTP	Network Termination Point
NSSK	North-South Station Keeping
NWM	Network Management
NWO	Network Operator
OAM	Operations Administration and Maintenance
OBP	On Board Processing
OMG	Object Management Group
OPEX	Operational Expenditure
ORB	Object Request Broker
OSI	Open Standards Interconnection
OSS	Operational Support Systems
PABX	Public Automatic Branch Exchange
PC	Personal Computer
PDF	Probability Distribution Function
pdf	Probability Density Function
PDH	Plesiochronous Digital Hierarchy
PDV	Packet Delay Variation
PIM	Protocol Independent Multicast
POTS	Plain old Telephone Service
PTT	Post, Telephony and Telecommunications
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying
RF	Radio Frequency

RFC	Request For Comments
RMI	Remote Method Invocation
RMTTP	Reliable Multicast Transport Protocol
RPC	Remote Procedure Call
RSVP	Resource reSerVation Protocol
RTD	Round Trip Delay
RTMC	Real Time Management Control
RTP	Real Time Protocol
SAW	Surface Acoustic Wave
SCPC	Single Channel Per Carrier
SDH	Synchronous Digital Hierarchy
SES	Society Europeenne des Satellites
SES	Satellite Earth Stations (An ETSI body)
SID	Shared Information and Data
SIP	Session Initialisation Protocol
SLA	Service Level Agreement
SN	Service Node
SNI	Service Node Interface
SNMP	Simple Network Management Protocol
SNWO	Satellite Network Operator
SOAP	Simple Object Access Protocol
SP	Service Provider
SQL	Specification Query Language
SSPA	Solid State Power Amplifier
STM(-N)	Standard Transport Module – (N depends upon rate)
SS-TDMA	Satellite Switched-TDMA
TCP/IP	Transmission Control Protocol / Internet Protocol
TDMA	Time Division Multiple Access
TINA	Telecommunications Information Networking Architecture
TMF	TeleManagement Forum
TMN	Telecommunications Management Network
TOM	Telecomms Operations Map
TT & C	Telemetry, Telecommand and Control
TTL	Time To Live
TWTA	Travelling Wave Tube Amplifier
UDDI	Universal Description, Discovery and Integration
UDP	User Datagram Protocol
UMTS	Universal Mobile Telephone System
UNI	User – Network Interface
VBR	Variable Bit Rate
VoD	Video on Demand
VPN	Virtual Private Network
VSAT	Very Small Aperture Antenna

WSDL	Web Services Description Language
WWW	World Wide Web
XML	Extensible Mark-up language
3GPP	3 rd Generation mobile Partnership Project

Publications related to this research

- M. Fitch. *The use of Satellite for Multimedia Communications*. BT Technology Journal, Volume 21, Number 3, July 2003. Pages 90 – 101.
- M. Fitch and A Fidler. *An overview of Satellite Access Networks*. BT Technology Journal, Volume 18, Number 3, July 2000. Pages 94 – 101.
- M. Fitch. *Operational Support Systems and QoS Management on Satellite Links*. IEE Seminar on End to End challenges of Broadband via Satellite. 26th January 2005.

Acknowledgements

First I would like to thank my external supervisor, Dr George Williamson, for his support, guidance and high level of commitment during this project.

I would also like to thank my internal supervisors, Professor Chris Bissell and Dr David Gorham for their interest, valuable feedback and guidance through the process.

Thanks are also due to my family and friends for enduring my moods during this project.

Finally I would like to offer thanks to my BT colleagues for the many corridor discussions and to BT as a Company for the use of laboratory, computing and library facilities.

1 Introduction

1.1 Background

Radio communications receivers and transmitters carried on satellites will always have a key advantage over ground-based (terrestrial) ones: they have a very high vantage point which gives them the ability to provide communications over large areas and long distances. For example, a geostationary earth orbit (GEO) satellite can cover about one third¹ of the earth's surface.

The high vantage point gives satellite links the following benefits over terrestrial links as pointed out by Fitch [1-1] and Cable [1-2]:

- the cost of providing communications is independent of distance,
- there are few nodes and no wires, leading to less maintenance and fewer outages due to manual errors and cable faults,
- signals can be broadcast / multicast to a very wide area that can contain any number of users and users can be quickly added or taken away without changes to the infrastructure,
- there is no sensitivity to country boundaries, hence they can provide reliable international communications, which is attractive to countries that do not have good communications infrastructure or that are landlocked with unfriendly neighbours,
- they are particularly suited to provision of services to remote, rural, aeronautical and maritime users where it is uneconomical or impossible to install terrestrial links.

These advantages are tempered by an unavoidable radio propagation delay, degradation due to bad weather when using some frequency bands, limited and expensive bandwidth and power, and a limited life. However each of these facets can be mitigated to a certain extent, this is discussed in chapter 2.

A major use of satellite capacity is television transmission. Satellite is now the most common method of delivery of subscription television services: in 2003, the number of household satellite TV receivers in the UK was 7.2 million, compared with 2.2 million cable TV installations [1-3]. Television broadcast is fundamentally a one-way service, but payloads have been launched on Astra 1H [1-4] and Eutelsat Hotbird 6 [1-5] satellites that are designed to support a return channel. The return channels are used for interactive TV and, to allow simultaneous receiving and transmitting from small cheap antennas, the carrier frequency of the return channel is very different from the forward channel. The return channel for these satellites uses Ka-band (about 30GHz), whereas the forward channel uses Ku-band (about 11GHz).

In addition to television services, satellites have been used since the late 1960s for two-way voice and data communications to maritime and aeronautical users and to countries that are

¹ Strictly, a geostationary satellite can 'see' almost half the earth's surface, but in practice the coverage is only one third because there is a limit to how low the earth station antennas can point (about 5 - 15 degrees above the horizon, depending on frequency).

not fibred; although decreasing in number, many countries still rely on satellite communications for telephone and data communications as pointed out in a speech this year by the CEO of the International Telecommunications Satellite Organisation (ITSO) [1-6]:

‘Today, for example, while more than 200 countries use satellite communications, 69 nations are using satellites as their primary (lifeline) technology for national and international telecommunications connections, and these countries comprise more than 60 percent of the world population’.

These international services usually comprise high data rate multiplexes (e.g. 120 Mbit/s) and use large earth stations of 10 – 20m diameter at Ku-band and C-band (4 – 6 GHz). Coming down somewhat in size, another important use is in Very Small Aperture Terminal (VSAT) networks where a corporation wants private communications between its offices located in different countries or in environments such as oil rigs and building sites. Typically VSATs have antennas that are around 1.8 to 3.4m in diameter. A recent trend is the growth of one-way and two-way satellite Internet access to premises that are out of range of copper systems like Asymmetric Digital Subscriber Line (ADSL)², enabled by the use of smaller and cheaper ground antennas, down to around 60cm diameter. The decline in use of satellite links for inter-country links and the growth of television and Internet services indicates that the use of satellite is migrating away from core network interconnection and towards the access network. The continuing growth of communications by satellite depends critically on the ability to succeed in this migration. The keys to success include a higher degree of integration with terrestrial networks and lower per-user costs, since the migration is from few high-value links to many low-value links.

The current high cost and difficulty in providing managed services over satellite means that telecommunications network operators prefer to maximise the use and reach of other technologies such as ADSL and WiMax³ when attempting to fulfil their promises of 100% broadband^{4,5} coverage. But despite the improvements in the reach of ADSL, there will always be some users that cannot be economically served by such technologies and satellite provides a solution for these, but the high cost means that the number of such users is quite low. According to an article in 2002 on the BBC News website, when describing the BT Openworld satellite service [1-7]:

‘There are more cons than pros with it compared with landline services. It is 400% more expensive to connect and half as fast [as ADSL]. On top of that, there is no return path other than by telephone line’

It seems that the attitude of large service providers is to use satellite for broadband access only as a last resort. It is a hope that this piece of research contributes a little bit to improving this situation.

² ADSL provides bandwidths of 256 kbit/s upstream and up to 1.5 Mbit/s downstream, depending on distance from the exchange. Currently the maximum distance is 6 km, where 512 kbit/s downstream is achievable.

³ WiMax is an emerging broadband wireless access technology based on IEEE 802.16 standard, with about 1 km range.

⁴ The definition of broadband is uncertain. It has been reducing over recent years and ADSL bandwidths are now commonly called ‘broadband’.

⁵ Service providers generally have to commit to providing services to an increasing percentage of the population over time, as part of their licence conditions. Some make this commitment public, e.g. BT in the UK.

1.2 The need for this research

Bearing in mind that a satellite link is only part of the end-to-end communications path, the various services carried by satellite require different degrees of integration with the terrestrial networks at each end of the satellite link. Services that demand higher Quality of Service (QoS) from the network, such as real-time video streaming and interactive voice, require greater integration to monitor and manage the QoS that is provided. Building on the Broadband ISDN model [1-8], such integration can be described in three planes: (1) the user plane where data is flowing, (2) the control plane where signalling is flowing for setting up and maintaining sessions and (3) the management plane for administering the network and services. The migration of use of satellite as described in the previous section is demanding more automation, more integration and lower per-user cost from all three planes as time progresses. Judging by the number of published papers, it is the management plane that has received the least research activity and yet it is at least as important as the others.

The Operational Support Systems (OSS) provide management plane support to services and networks. A library search on Satellite OSS revealed fewer than 10 papers and these are mainly on future satellite systems. From the research point of view, it is nearly a green-field site.

Further pressure to increase the automation and integration in OSS is coming from changes in the wider telecommunications business environment. The history of OSS only goes back about 40 years, about the same as satellite communications. The first systems were often fragmented and specific to network and service type. They were first developed by large telecommunications companies that were largely government owned and had limited competition. Since then, privatisation and de-regulation in the 1980s and 90s have spurred an increase in the number of companies offering telecommunications services, competing with the original telecommunications companies. In this new environment, automation and integration of OSS is essential to reduce costs and increasing speed and reliability of end-to-end service provision and hence maintain a competitive edge. OSS business frameworks, architectures and technology have been developed through a number of standards and industry bodies to the point where software products and platforms are available off-the-shelf.

1.3 The research questions

When considering the development of satellite systems and OSS, two questions have been framed on how OSS developments can enable satellite services to be more competitive, which are:

- what OSS processes are different for satellite and why are they different ?
- are there particular processes, particularly suited to satellite communications systems, that can be developed further in order to assist them to become more competitive ?

At this point it is necessary to state that the satellite links are assumed to be carrying packet data. Internet Protocol (IP) packets are used in the modelling and simulation activities in this study; common IP packet lengths are used with traffic sources that are modelled on Internet traffic. However no layer 3 (network layer) functionality is assumed so the results are easily

extended to frame-based and cell-based links (such as Asynchronous Transfer Mode (ATM) links).

The answer to the first question is that there are several differences but the primary ones are in the area of performance management, in particular Quality of Service (QoS) management. The differences are mostly at layer 1 (physical layer) of the Open Systems Interconnection (OSI) model, since this is where the majority of satellites operate. The reasons stem from the characteristics of satellite links: they cover a large geographical area, have a high latency, have hard capacity and power limits, and are expensive.

To answer the second question, the management of QoS is split into three parts: measurement, process and action. The focus is mainly on the first of these. Measurement of QoS parameters has to be performed in real-time, while creating the minimum amount of overhead traffic and at the same time requiring low processing power at the remote networks, since these are likely to be small and located at user premises. The use of small trace packets is investigated to estimate packet delay and loss and the limits of validity of this estimation method are evaluated. It is similar to the use of Operation, Administration and Maintenance (OAM) cells used in ATM to measure QoS parameters including delay and loss of user cells as described in ITU-T recommendation I.356 [1-9], but extends the idea for use with IP packets. The trace packet method is compared with other techniques such as direct packet sampling and is shown to have advantages in terms of overhead traffic and required processing power, especially when the satellite link is congested.

These results support the argument for a position as follows:

- that OSS for satellite links are different to that for terrestrial links particularly in the area of management of QoS parameters of delay and loss of data,
- the reasons for the differences include the migration in the use of satellite towards the access network, larger latency caused by radio-wave propagation delay, and limited and expensive transmission capacity,
- that estimation of delay and loss of data can be performed by using small trace packets that, in conditions of high load, has advantages over direct sampling methods in terms of overhead traffic and required processing power.

The new results presented in this study are discussed in the context of a management framework and used to propose management modes and some simple guidelines to assist automation and integration.

1.4 Structure of the thesis

Figure 1-1 shows the overall structure of this study, expressed in the form of chapter titles and information flows.

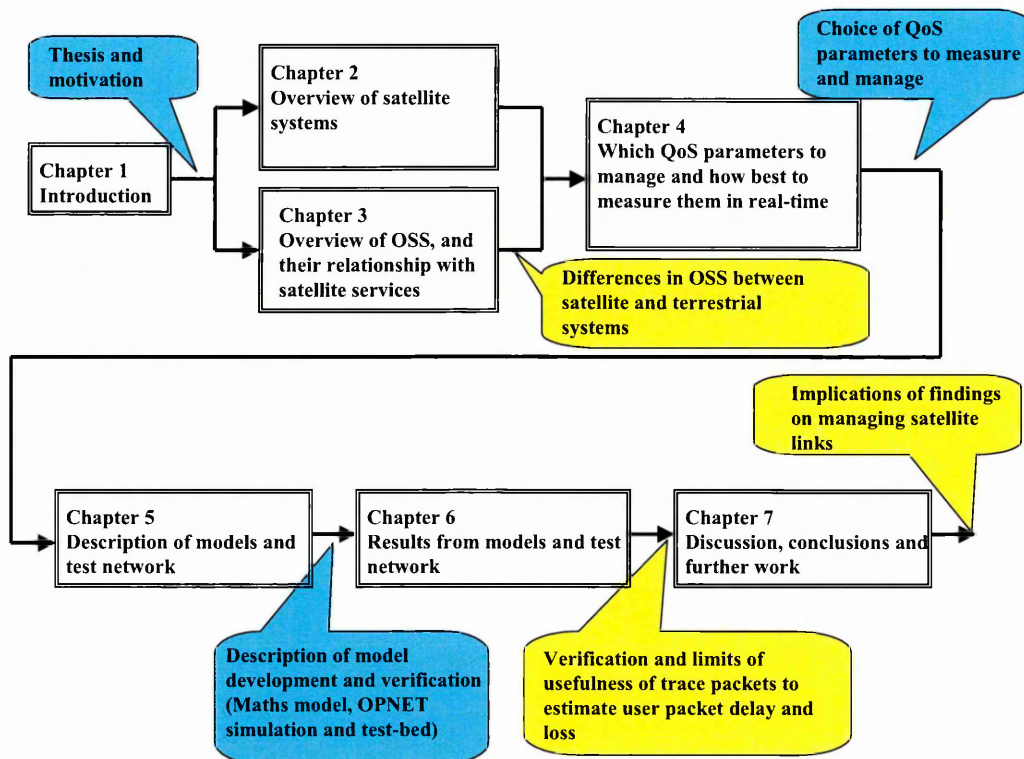


Figure 1-1 Structure of the study

Chapter 2 is an overview of satellite systems containing descriptions of satellites, orbits, access schemes, link types and interfacing to networks. The chapter ends with an appraisal of changing market and business conditions facing satellite operators.

Chapter 3 is an overview of OSS and contains an account of some recent research activity relevant to satellite systems and also includes a few examples of practical implementations. In this chapter is a re-thinking of the commercial value chain and an analysis of the enterprise Telecoms Operations Map (eTOM) business process framework, to derive the differences between satellite and terrestrial OSS processes.

Chapter 4 focuses onto the QoS parameters that are of most relevance to satellite links. It draws heavily on standards activities, since open standards have been established as the only way forward when procuring products from different vendors with the expectation that they will work together. This chapter contains a mapping of services to QoS parameters and an overview of the commercially available tools and common methods to measure these parameters. Finally, the chapter introduces and discusses trace packets as a method of measurement.

Chapter 5 is a description of the models used in this study. Included are traffic models and queue models with mathematical descriptions, the OPNET modeling tool and how the

simulation models were built onto it and calibrated, and a description of a practical test network.

Chapter 6 presents the results of the OPNET simulations and measurements performed on a test network and a live satellite link. The results enable measurement modes to be discussed where QoS parameter estimation methods can usefully be varied as a function of network load.

The original work in this thesis starts during chapter 3 with the analysis of the eTOM and flows through chapters 4, 5, and 6, where chapter 6 contains the majority of the novel results. A particularly novel proposal from this chapter is to measure the variance in the delay of trace packets as they arrive at the remote network and use this, along with the estimated probability density function (pdf) of the delay, to determine the percentiles or quantiles of the delay distribution as recommended by the relevant standards I.380 (now re-numbered Y.1540) [1-10] and Y.1541 [1-11]. The variance is also used to estimate traffic load on the link and hence provide information to decide when to switch between estimation methods.

Some of the above chapters end with a summary of the findings in the chapter to make the thesis easier to read. Chapter 7 is a short chapter that draws together the main conclusions and proposals for further work.

1.5 References for chapter 1

- [1-1] M Fitch. *The use of Satellite for Multimedia Communications*. BT Technology Journal, Volume 21, Number 3, July 2003. Pages 90 – 101.
- [1-2] Stephen Cable. *Network Bandwidth on Demand for Multimedia and Internet Applications*. Satellite Communications magazine, May 1999. Pages 48 – 53.
- [1-3] Euromonitor International, October 2004, executive summary. See [http://www.euromonitor.com/Cable and satellite TV services in UK \(mmp\)](http://www.euromonitor.com/Cable_and_satellite_TV_services_in_UK_(mmp)). Link checked 11 November 2004.
- [1-4] SpaceDaily article January 28th 2003 on Astra 1H. See <http://www.spacedaily.com/news/satellite-biz-03g.html>. Link checked 11 November 2004.
- [1-5] Eutelsat: *The Hotbird satellites*. See http://www.eutelsat.org/satellites/4_2.html. Link checked 11 November 2004.
- [1-6] Ahmed Toumi. *Global Broadband Satellite Infrastructure Initiative*. Presentation to the European Institute Roundtable on Telecommunications, E-Commerce and Audiovisual Policies, 25th February 2004. See <http://216.119.123.56/dyn4000/dyn/docs/ITSO/broadband/broadband2004/EDITED%20European%20Inst%2025%20Feb%2004%20Prsentation.htm>. Link checked 19th November 2004.
- [1-7] *Lift-off for low-cost satellite broadband*. BBC News website 15th April 2002. See <http://news.bbc.co.uk/1/hi/sci/tech/1931217.stm>. Link checked 19th November 2004
- [1-8] S Bhatti. *B-ISDN Protocol Reference Model*. University College London website, March 1995. See <http://www.cs.ucl.ac.uk/staff/S.Bhatti/D51-notes/node39.html>. Link checked 28 November 2004.
- [1-9] *B-ISDN ATM layer cell transfer performance*. ITU-T recommendation I.356, version 10/96 October 1996.
- [1-10] *Internet Protocol data communication service – IP packet transfer and availability performance parameters*. ITU-T recommendation I.380, version 02/99, February 1999.
- [1-11] *Network Performance Objectives for IP-based Services*. ITU-T Recommendation Y.1541, May 2002

2 Overview of satellite systems

2.1 Introduction

This chapter gives an overview of satellite systems, what satellites are, the services they provide and how the technology and market-place is changing with time. The chapter is arranged as follows. Section 2.2 gives an overview of what satellites are, sections 2.3 and 2.4 describe what they are good at and what their limitations are. Section 2.5 gives a technical overview of satellite systems, the radio spectrum they use, the various kinds of orbit and typical services they provide. Section 2.6 describes how the market-place is changing and the response to this change from the satellite operators. Finally section 2.7 gives some examples of commercial satellite systems.

Although this chapter is not primarily about OSS (this is introduced in the next chapter), where an issue or development of satellite systems has implications on the OSS, this is highlighted.

2.2 What satellites are

There are many kinds of satellite for many kinds of uses, such as earth observation, space observation, geo-location and weather prediction and communications. The subject of this study is limited to communications satellites.

Not only are there different kinds of satellite, but each kind can be operated in one of several different kinds of orbit. This study will focus mostly on the GEO¹, since this is by far the most common orbit for communications satellites. There have been attempts to use other orbits that have met with mixed success; an account of these is included in the technical overview in section 2.4.4.

Communications satellites are made up of two basic parts, called the bus and the payload. The bus refers to the mechanical structure of the satellite, its power and thermal sub-systems and its fuel tanks and propulsion system. The payload refers to the electronics that handles the communications signals, including the antennas. It is the payload that is of interest in this study.

The history of satellites goes back only about 50 years. Maral and Bousquet [2-1] point out that the basic technologies used in satellites, missiles and microwaves, were developed largely in the second world war. The first satellite launched was Sputnik in 1957, followed by various experimental satellites such as Courier (1960) that could store and forward transmissions and Telstar (1962) which had wide-band repeaters. The first commercial satellite was Intelsat 1 (or Early Bird) that had a capacity of 480 telephone channels. Intelsat 1 mass was 68 kg at launch

¹ The GEO is circular with an altitude of around 36000km and an orbit period equal to the rotation period of the earth. Satellites are positioned above the equator and therefore appear to remain in one spot in the sky from an observer on the earth's surface. In practice there is usually a slight daily movement caused by imperfect station-keeping.

in April 1965 and it had a lifetime of 1.5 years. The cost per telephone channel was \$30,000 per year (at that time).

Since then, satellite and launcher technology has developed to allow much heavier, more powerful and longer-lasting satellites to be placed in service. A recent example is Intelsat 10-02, built using the Eurostar E3000 bus. It had a wet (fuelled) mass of 5500 kg, a power budget of 11 kW, many thousands of equivalent telephone channels and a planned lifetime of 13 years when launched in June 2004 [2-2]. The cost per telephone channel is now less than \$400 per year to service providers. Increasing capability in microwave technology has enabled satellites like Intelsat 10 to have contoured multi-beam antennas whose beams adapt to the shape of continents.

2.3 Satellite link characteristics

2.3.1 Wide area coverage

Repeated from chapter 1, satellite links are characterised by wide-area coverage which carries the following benefits:

- the cost of providing communications is independent of distance,
- there are few nodes and no wires, leading to less maintenance and fewer outages due to manual errors and cable faults,
- signals can be broadcast / multicast to a very wide area that can contain any number of users and users can be quickly added or taken away without changes to the infrastructure
- there is no sensitivity to country boundaries, hence they can provide reliable international communications, especially to countries that do not have good communications infrastructure or that are landlocked and have unfriendly neighbours,
- they are particularly suited to provision of services to remote, rural, aeronautical and maritime users where it is uneconomical or impossible to install terrestrial links.

2.3.2 Typical transmission rates

The transmission rates on a satellite link vary typically between 64 kbit/s and 155 Mbit/s (which is the current state of the art limit) depending on the service. Commercially there are many more lower bit-rate links than there are higher ones; 128 kbit/s, 512 kbit/s and 2 Mbit/s are the most common sizes for communications and Internet access to small terminals. DVB-S television multiplexes are typically 35 Mbit/s and carry around 10 channels each. A modern satellite can support many such links and the total maximum capacity for a whole satellite can be more than 1 Gbit/s.

2.3.3 Propagation delay

Delay associated with satellite links has given them a bad reputation in the past, especially on voice circuits with inadequate echo cancellation, which is again becoming an issue with VoIP even on terrestrial links. However, if satellites are used for data or voice with modern echo cancellation techniques, they are entirely acceptable for all types of services except perhaps highly interactive ones like games.

The radio-wave propagation delay (due to satellite-earth distance) is typically 240 – 280 ms per hop for a GEO satellite, depending on location of earth stations and satellite orbital

position. A round-trip delay would be double this (two hops). Delays for other types of orbit are discussed in section 2.4.4 on orbit types. Although the GEO delay is high compared to delays on voice circuits around the world, it is not high when compared to delays on the Internet. Puetz [2-5] has published a comparison of satellite and terrestrial network round-trip times he experienced on Internet connections from New York to other cities around the world, reporting between 540 and 918 ms for a GEO satellite link and 305 ms to 3.3 seconds for terrestrial links that went via many router hops. As he points out:

‘a satellite link only yields a 292 ms one-way delay on average with very little difference between the longest and shortest ping times’

The reason that Puetz measures an average 292 ms over satellite instead of 260 ms as implied in the above paragraph is most likely due to additional delays caused by error-correction coding and buffering in satellite modems.

2.3.4 Outages due to bad weather

The weather affects signals to and from satellites to a varying extent, depending largely on the signal frequency. As a rule, the higher the frequency, the worse the signal is affected.

The most serious effect is attenuation caused by rainfall, the periods of attenuation caused by this are termed ‘rain fades’, which can be 20 dB or more deep at the higher frequencies. Scintillation is another, less serious effect caused by thermals in the atmosphere and also by edges of clouds, whereby the signal can rise and fall in level by 1 - 2 dB, again dependent on frequency.

As a safeguard against these fades, satellite links are generally operated with a power margin that is usually called the ‘clear sky’ margin. This margin can be fixed, i.e. there all the time, or it can be invoked when needed. The common methods of altering the power margin are power control and adaptive coding and modulation. Both have the effect of altering the transmitted energy per user bit period.

Both of these methods require signalling across the link to agree and synchronise changes at each earth station. This adds complexity, cost and signalling overhead and such schemes have been implemented in only a few cases, typically high bandwidth trunk links with large earth stations. For these reasons, a fixed margin is normally used for smaller earth stations (which are more relevant to this study) where the power, modulation depth and coding scheme are all kept constant. A typical clear sky margin of 6 dB is used on commercial satellite services, which is adequate to cope with most rain fades up to Ku band (11-14 GHz), which currently contain the majority of services to smaller earth stations.

However, at higher frequencies such as Ka and V band (20 – 50 GHz), rain fades can be 20 dB or more deep and these fades cannot be mitigated by power control or by changes in modulation and coding schemes even if present. Therefore the link suffers complete outages due to rain fades and during these outages the link is said to be ‘unavailable’, whereas for the remainder of the time the link is ‘available’. Such fades can last for a few seconds up to a few minutes depending on the size of the rain cell. Work done at the University of Bath shows that fades at Ka and V band are characterised by short deep fades [2-6]. The ITU have continued

this work to derive some preliminary models for fade durations [2-7] and the COST280 project is working on fade mitigation techniques for these bands [2-8].

The ITU-T has standard rain models, against frequency, for specified rain regions around the world; from these models it is possible to look up availability figures against link margin for annual average and worst month [2-9]. The worst month is typically 0.2% worse than the annual average. The satellite network operators deal with this situation by building availability into the Service Level Agreements (SLA), for example by guaranteeing 99.7% availability averaged over a year, with 99.5% over the worst month.

2.3.5 Limited power and bandwidth

Currently transmitters on board satellites are capable of generating around 100 W per beam. The limited power and greater range means that larger antennas must be used at earth stations than with terrestrial radio links of the same capacity. The highest data rate that can be transmitted over a satellite link is currently 155 Mbit/s over a standard 72 MHz-wide satellite channel using, for example, modems made by Newtec [2-3].

2.3.6 High cost and limited life

Satellites typically cost \$125m - \$250m to manufacture (depending on complexity), plus the same amount again for launch and the same amount again for insurance. This high cost means that between 70 and 90% of the total cost of providing a satellite service is incurred by the space segment, depending on the bit-rate [2-10]. The remainder of the cost is made up from earth stations, their connections to networks (the backhaul) and other costs associated with providing the service.

The high cost of insurance is caused by a high probability of launch failure, estimated by Bear Stearns to be about 0.1 [2-11] and the history of satellites is peppered with such failures. However, once launched successfully, the probability of the satellite failing in orbit is very much lower and in fact satellites generally exceed their design life. Some GEO satellites which were designed for 10 year lives are still in orbit after 15 years, albeit usually with some loss of payload redundancy and with some orbit inclination. In contrast, low earth orbit (LEO) satellites generally last around 7 – 10 years. The life-time limit for satellites is usually set by the amount of fuel it has remaining after it arrives at its orbital location and, for this reason, GEO satellites are best launched from the equator, since they then do not have to dump any angular momentum from the earth. There will be an optimum longitude for launch, which cannot always be satisfied by the few equatorial launch sites, such as Kourou. This is the motivation behind Boeing Sea Launch; Boeing have adapted an old oil rig as a launch pad and can tow it to the equator at a suitable longitude, which will enable satellites to have the maximum fuel remaining once in position. Satellites cannot be modified, repaired or retrieved once launched. Therefore they should be as simple and reliable as possible to maximise their useful life.

2.4 Technical overview of satellite systems

2.4.1 Frequency bands

Satellites use radio frequency bands in the microwave region. The spectrum allocation (i.e. the frequency bands that satellites use) is co-ordinated world-wide by the International

Telecommunications Union (ITU)[2-12]. Table 2-1 contains some common satellite operating frequencies and some typical services that they support.

Frequency band	Frequency range	Typical Service	Typical data rates	Example satellite operator
L-band	1.5 – 1.6 GHz	Voice and data to mobile terminals	2.4 – 64 kbit/s	Inmarsat
S-band	2.3 – 2.5 GHz	Voice and data to mobile and hand-held terminals	2.4 – 9.6 kbit/s	Iridium
C-band	4 – 6 GHz	Fixed services, international trunks, telephone and leased circuits	64 kbit/s – 120 Mbit/s	Intelsat
X-band	8 GHz	Military communications	2.4 - 256 kbit/s	Skynet 4 / 5 Nato 3
Ku-band	11 – 14 GHz	Television, VSAT networks, international data	128 kbit./s – 155 Mbit/s	Intelsat, Eutelsat, Astra
Ka-band	20 – 30 GHz	Television back-channel, Satellite Broadband	64 kbit/s – 2 Mbit/s	Astra (TV), Spaceway (Broadband)

Table 2-1 Common satellite frequency bands and example services

In table 2-1 column 2, most bands are divided into two sections, for space-to-earth and earth-to-space, with the latter at the higher frequency. The exceptions are the L-band and S-band allocations which are both entirely space-to-earth. Satellites that operate in these bands use C-band for the uplink.

2.4.2 Transparent satellites

The vast majority of satellites, past and present, are ‘transparent’, that is, they do not demodulate the signals as they pass through the satellite. Some satellites switch the signal between beams according to a fixed or changing pattern and they can filter the signal (using either analogue or digital means) but they are still classified as transparent because they do not perform demodulation. Transparent satellites are sometimes referred to as ‘bent-pipe’, since they operate only at layer 1 (physical) of the OSI 7-layer model.

These satellites receive signals from earth in an antenna beam, translate the carrier frequency of the signals from the uplink frequency band to the downlink frequency band and transmit the signals back in a different antenna beam (which may or may not overlap geographically with the uplink beam). This frequency-changing operation is sometimes called ‘transponding’ and hence the electronics in the satellite payload that handles the communications signals through a channel is called a transponder.

Transponding the signal means that any Doppler shifts, noise and interference on the uplink are translated to the downlink along with the original signal. It also means that the downlink power is a function of the uplink power, so that if the uplink suffers attenuation due to, say, heavy rainfall, the downlink power also drops. Ways around this include fade margin and power control as discussed in section 2.3.4.

The signals, which are radio-frequency (RF) modulated by appropriately coded and framed base-band data, are referred to as ‘carriers’ in the normal radio sense.

Within the overall spectrum allocation, band-pass filters on board the satellites divide up the bands into 'channels', each handled by a different transponder, so that several channels are carried over one satellite. As an example, the Astra 1C digital TV broadcast has 18 Ku-band transponders [2-4], each 36MHz wide and each carrying a multiplex conforming to the Digital Video Broadcasting - Satellite (DVB-S) standard [2-13]. Each multiplex contains around 10 TV programmes, thus a single satellite can broadcast typically 180 TV programmes.

Other examples of transparent satellites are:

- the Intelsat fleet used for fixed earth station communications, which generally have transponders 72 MHz wide at C-band and 36 MHz wide at Ku-band,
- Skynet 4 used for military communications with transponders up to 8 MHz wide, hardened against interference at X-band,
- Eutelsat used for TV broadcast and some mobile services with transponders 36 MHz wide at Ku-band,
- the Inmarsat fleet used for aeronautical, maritime and land mobile communications, with transponders up to 100 MHz wide, with C-band uplink and L-band downlink.

The Inmarsat 4 satellites represent the state of the art in on-board digital filtering. Due for launch in the first half of 2005, they have a very sophisticated payload that uses digital filtering and switching between beams, which allows them to vary the transponder bandwidth of a few kHz up to a few 10s of kHz. The advantage of doing this is that the processor can be combined with narrow spot-beams so that the scarce L-band frequency spectrum can be re-used many times from the same satellite. Inmarsat 4 satellites are classified as transparent because the payload does not demodulate the signals, only filters them.

Figure 2-1 shows the essential elements of a transparent satellite communications payload.

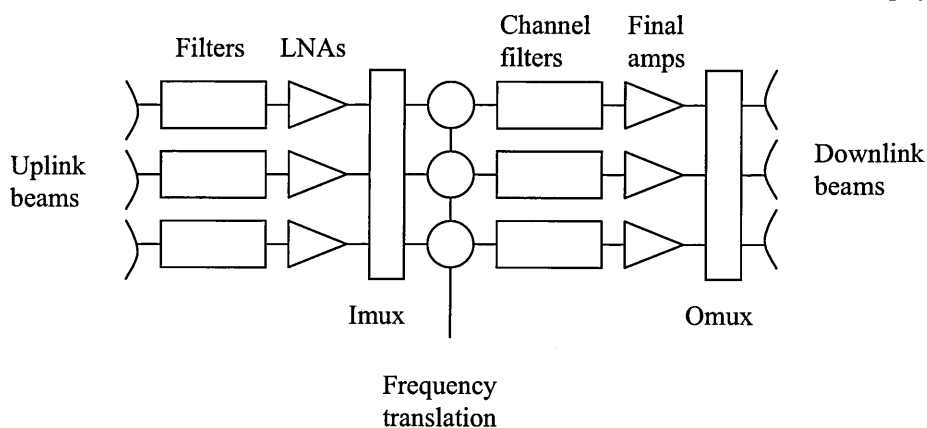


Figure 2-1 A simple generic transparent payload

Moving from left to right on figure 2-1, the signal is received from the ground in the uplink beams, filtered (fairly coarsely) and amplified using low noise amplifiers (LNAs). The Imux (Input mux) separates out the channels from the beams and re-forms them into different combinations (if necessary) for the downlink beams. The frequency translation stage converts the uplink frequency to the downlink frequency by mixing with a local oscillator. The channel filters are narrow, often employing digital filters or surface acoustic wave (SAW) technology.

Following the filters, the signal is amplified to a high power in the final amplifiers and fed to the downlink beams via the Omux (output mux) that performs the dual tasks of isolating the amplifiers from one another and filtering the signal to the output antennas. In general, there are more final amplifiers than there are downlink beams, to give a level of redundancy for reliability purposes.

There is a maximum transmit power associated with satellite down-link beam, set by the size of the final amplifier in the chain. As stated above, the combination of input amplifier, channel filter, frequency translation and final amplifier (one horizontal path through figure 2-1) is referred to as a ‘transponder’. A transponder final amplifier will be either a travelling-wave tube amplifier (TWTa) or a solid-state power amplifier (SSPA) and powers of 100 W are typical.

Bringing the last two points together, each transponder on the satellite has a maximum power and a maximum bandwidth, which are together referred to as the transponder ‘resource’. With transparent satellites it is possible to lease a part or all of a transponder from a satellite operator. If the proportion leased is ‘ x ’ (where $0 < x \leq 1$) then the user is considered to have taken x amount of resource when he has used up either x amount of the power or x amount of the bandwidth, whichever occurs sooner. The transponder resource can be power limited or bandwidth limited and the most efficient use of the transponder is achieved when the carrier planning is such that the power and bandwidth limits are reached simultaneously.

2.4.3 Processing satellites

This second major satellite type performs demodulation on the signals to recover the baseband data and then performs switching or routing of the baseband signals to different downlink beams. These satellites are so-called OBP (on-board processing), sometimes they are referred to as ‘regenerative’. They are much more complicated than transparent satellites.

Figure 2-2 shows the essential elements of a processing satellite, taken from a paper by TRW [2-15]. Please note is that this diagram is a simple example only.

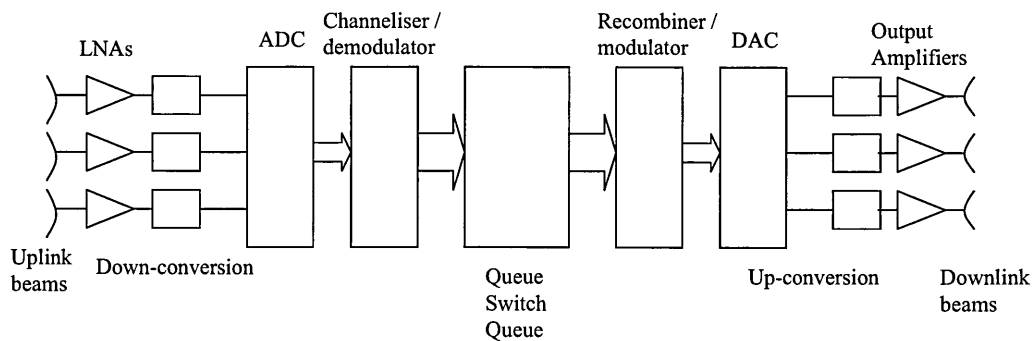


Figure 2-2 A simple generic processing payload

The essential elements of a processing satellite, over and above transparent satellites, are the demodulation, queuing- switching stages and re-modulation.

A good overview of the technologies for processing payloads is given by Petz et al [2-14], who describe the necessary characteristics and state of the art for low-noise amplifiers, down-converters, demodulators and cell switches. They conclude that the acceptance of OBP systems depends upon the service quality, cost of the user terminal and, interestingly, the inter-operability with terrestrial systems.

OBP satellites have the following characteristics:

- they de-couple the uplink from the downlink, hence they do not translate Doppler shift, neither do they re-transmit noise or interference. They are therefore also more resilient to jamming signals or bit errors that manifest at the satellite receiver, since forward-error correction can be incorporated on-board,
- they are fitted with multiple spot-beam antennas and hence can re-use the RF spectrum many times, without the requirement that gateway earth stations be present in every beam. For example Spaceway satellites have antennas comprising some 1200 horns that are coupled with a beam-forming network in order to form 40 beams, each served from an on-board switch. With a re-use pattern of 3, the spectrum is re-used over 10 times [2-19],
- the scheduler which manages access by earth stations is on-board and can be tightly coupled to the on-board queues without the high latency experienced with transparent satellites, hence congestion is minimised on the satellite link and pushed out to the terrestrial network as much as possible,
- they present a bigger OSS challenge than transparent ones since there are more elements on board to manage elements such as demodulators and switches. They are also a shared resource, which needs to be managed, and the way in which the bandwidth on the satellite is divided among service providers is not a trivial problem. It is not as straight-forward as leasing transponders on a transparent satellite, but requires the allocation of reserved and pre-emptible capacity. They also have non-independent queues before and after switches, and management of these queues places a heavy load on the on-board processor.

The major technical challenges with OBP satellites are the power consumption and size / mass of the demodulators, processor power and reliability of switches and antenna beamforming networks.

During the late 1990s there was a large emphasis on 'broadband' satellite systems and at least three US systems were designed (but none were launched), utilising very large and powerful OBP satellites at Ka-band, where there is sufficient bandwidth available to support broadband services. These three were Lockheed Martin's Astrolink, the Intelsat Broadband system and Spaceway (begun by Hughes who were taken over by Boeing and then sold to News Corporation). There has also been a European development: the European Space Agency (ESA) has part-funded the development of EuroSkyWay (not yet launched), which is also a broadband satellite system with OBP that will operate in Ka-band.

Astrolink was based on ATM and the payload contained an ATM switch that was adapted from a Motorola design originally intended for military satellite use. Astrolink was abandoned for various reasons, one being mass and power problems with the ATM switch. Regarding the Intelsat Broadband System, Intelsat looked very thoroughly into the issues, including commercial considerations and dimensioning the system [2-16] and giving presentations e.g. [2-17], before privatisation made it too risky for them to fund from debt (more on this later in

the chapter). The feeling in the industry at that time was that if Intelsat, with its many years of experience with satellite operations, was not willing to invest in a broadband system, then doubt was cast on other providers to make such a commercial success of such a system.

As for Spaceway, three satellites were built and, up until late 2004, it looked as if the first two satellites would be launched, but News Corporation have decided to scale back the project to launch just one satellite to deliver High Definition TV programming to North America, rather than broadband mesh services. The new launch date has yet to be announced.

Of all the broadband OBP satellite initiatives, only EuroSkyWay survives but the launch date is not known. EuroSkyWay is partially funded by the European Space Agency (ESA) and is targeted at the European, Middle East and African markets; a description is given by Spagnolo et al [2-18].

One OBP system that has been a commercial success is 'Skyplex', which is a development funded by ESA and flown on the Eutelsat Hotbird 5 satellite. It supports the very specialist television contribution market: this satellite has several receivers on-board which demodulate uplinked TV signals; these are then combined to form a single downlink multiplex [2-20]. The benefit of this system is that several earth stations can contribute TV feeds into the downlink multiplex.

Non-broadband examples of OBP satellite systems are Iridium and Globalstar, which are both LEO systems, built respectively by Motorola and Alcatel. These systems are aimed at the voice and low-rate data market rather than data communications, which means that they are circuit oriented rather than packet oriented and their OSS requirements are different. They use low-rate voice codecs and short fixed-length on-board cell switches. They have not been commercially successful, due mainly to competition from the unexpectedly fast expansion of GSM networks across the world, logistic problems with handset availability and OSS problems with signing up customers quickly with reliable billing.

The OBP satellites discussed so far perform switching at the link layer, using fixed short-length cells and high-speed hardware switches, for reasons of maximum flexibility and longer useful life. Only one case is currently known of a satellite operating at network layer (layer 3): Cisco has mounted a router on a small satellite manufactured by the Surrey Space Centre [2-21]. But a satellite that performs routing in addition to link layer switching is more complex, less flexible and requires additional overhead such as reliable uplinking of routing tables.

Since there will be only very few OBP satellite systems operating commercially in the foreseeable future, the major emphasis in this study will be upon transparent satellites. OBP types are mentioned occasionally for completeness.

2.4.4 Orbit type

Either type of satellite, transparent or OBP, may be placed in any orbit. The three most common circular orbits are:

- Low Earth Orbit (LEO) - up to about 1500 km altitude, with an orbital period of about 90 minutes. These have a propagation delay of around 10 ms per hop. In fact they have less

delay than a fibre link if the path distance along the ground is more than about 2,000 km since the speed of light in fibre is only about $0.6c$. A constellation of many 10s of satellites is required to cover the earth's surface.

- Medium Earth Orbit (MEO) - up to about 15,000 km altitude, with an orbital period of about 10 hours. The propagation delay is around 100 ms per hop. A constellation of about 10 satellites is required to cover the earth's surface, spread out on orbits equally spaced in inclination.
- Geostationary Earth Orbit (GEO) – at 32768 km altitude, with an orbital period of 1 day. The propagation delay for signals is nominally $3.28 \times 10^7 / c = 0.11$ seconds where c is the speed of light ($= 3 \times 10^8$ m/s). So the delay for one 'hop' would be a minimum of 220 ms, but this would only be possible if the signal were transmitted and received by the same earth station situated on the equator directly beneath the satellite. Three satellites are required to cover the earth's surface, equally spaced in longitude with ideally zero inclination.

Practically, the time delay for a hop between two separate earth stations coupled by a GEO satellite is usually quoted as between 240 – 280 ms. This variation is caused by different slant path lengths, which in turn depend on the difference in longitude between the satellite and the earth stations and on the latitude of the earth stations. This delay is increased by further small delays (of about 50 – 80 ms) added by the modems at either end of the link, caused by processes such as buffering and forward error-correction coding (FEC). This means that the round-trip delay (RTD) for a TCP / IP session that includes a satellite link is typically 630ms for the satellite hop alone (assuming both way by satellite).

Some orbits repeat the same ground track, these are synchronised with the earth's rotation and are said to be in geosynchronous earth orbit (GSO). GEO is always GSO, whereas LEO or MEO may or may not be.

LEO and MEO satellites orbit the earth in less than 24 hours and therefore they move across the sky when observed from the earth's surface; in the case of Iridium the satellites can be seen for about 10 – 20 minutes depending on the user position. This means that earth stations or hand-held terminals must either have omni-directional antennas or, if they have directional antennas, they have to track the satellites across the sky. Two such antennas are required for continuous service, since one satellite will be setting over one horizon while the next one will be rising over a different horizon.

The most common type of circular orbit is GEO, with its orbital period of 24 hours. Ideally, satellites in this orbit hang in the sky in one position and so earth station pointing can be fixed. In practice, even GEO satellites move slightly and trace out a figure '8' in the sky whose dimensions depend on how accurately the satellite is on station. When launched, these satellites sometimes have some inclination deliberately introduced, so that the '8' is 2 – 3 degrees north-south (N-S) (peak-to-peak) and about one-tenth of this east-west (E-W). During the life of the satellite, the N-S movement will decay to zero and build up in the opposite direction, reaching its original size after a few years. The N-S station-keeping is much more expensive on fuel than E-W (the reason is due to changes in angular momentum), so the satellite operator performs the minimum of manoeuvres in this direction. Small earth stations generally do not need to track GEO satellites, since they have a beamwidth of typically 10 degrees, but larger gateway antennas (say 20m diameter) generally do need to track because

they have beamwidths of < 1 degree. The size threshold beyond which tracking becomes necessary depends both on the operating frequency and the size of the antenna; the beamwidth is inversely proportional to both the frequency and the area of the antenna.

In addition to the circular orbits, there are highly elliptical and highly inclined orbit types such as the Molnya, Tundra and Loopus which are designed to provide mobile communications to high latitudes. Some of these are also GSO and it is not always necessary for earth stations to track the satellites, since they can spend several hours not changing direction from the earth station (although they change greatly in range). For example, 3 satellites in the Molnya orbit can provide uninterrupted coverage over much of Russia. The propagation delay to these orbits can be higher than for the GEO orbit.

More detailed information on orbit types and propagation delay, including derivations of orbit mechanics, can be found in Maral and Bousquet [2-22].

2.4.5 Number of satellites required

GEO satellites are positioned in what is known as orbital 'slots', that are 2 degrees apart and therefore 180 slots exist, although some slots are more valuable than others; for instance those over the Indian Ocean are all high value and all occupied, to carry traffic between Europe and Asia. The slots are allocated and managed by the ITU and strict rules apply to filing for slots and frequency co-ordination, since they are spaced too closely to allow the smaller earth stations to discriminate between them by their beam patterns (a VSAT may have a beam diameter of 10 degrees).

The fact that there are only 180 slots does not limit the number of GEO satellites to 180; some operators put more than one satellite in a slot to provide more bandwidth. Examples are SES who have seven Astra satellites (flights 1B, 1C, 1E, 1F, 1G, 1H, 1K) in one slot at 19.2 degrees East for TV direct-to-home services in Europe and Spaceway who were planning to place four satellites in one of its slots and three in another.

The lower the satellites are in their orbits, the more of them are needed to provide global coverage. It is possible to cover the earth's entire surface from 3 GEO slots (with the exception of the poles), while with MEO it takes about 12 and with LEO it takes 40 or more. So the up-front cost of introducing services from MEO and LEO systems is very high, because most of the constellation has to be present before service can start. Also, with MEO and LEO systems, much of the capacity is always out at sea and, in the case of Iridium, over the poles.

2.4.6 Telemetry, Telecommand and Control

The TT & C system comprises a sub-system on the ground and one on the satellite. The purpose of the TT & C system is to fly and manage the satellite and it must therefore have connections to the bus and payload portions.

The TT & C earth stations are separate from the communications earth stations and are run by the satellite operator. Usually at least two TT & C earth stations are used to manage satellites (main and standby) and many more are used during the launch phase. The radio channels used by TT & C are usually different from those used for the communications channels, although

the US FCC is trying to force satellite operators to align them. C-band is the most popular for TT & C because of its high reliability in bad weather.

The satellite payload will have receivers, transmitters and antennas dedicated to TT & C with data connections to the bus to manage the power subsystem, thermal subsystem, thrusters and fuel subsystem. The payload itself will also be managed by TT & C, with a range of commands and parameters appropriate to a transparent or processing payload.

Using the TT & C, the satellite becomes another set of elements to be managed that requires integrating with the rest of the OSS. However, since the management traffic is separate from the communications traffic and hence does not add overhead traffic, it is not considered further in this study.

2.4.7 Multiple access by user terminals

Multiple access means several users can access the satellite simultaneously, with transmissions separated by frequency-slots, time-slots, a mixture of the time and frequency slots, or different pseudo-random codes. There follows a brief description of these main types of multiple access, which are respectively FDMA, TDMA, MF-TDMA and CDMA.

FDMA

Frequency Division Multiple Access (FDMA) is where each user is allocated a different frequency at which to transmit. This is sometimes known as Single channel per carrier (SCPC) and is used mainly for leased circuits where bandwidth is constantly reserved for each user, whether or not it is being used. These are rarely used since they are expensive and do not efficiently utilise the satellite capacity (since no other users can use the capacity during quiet periods).

TDMA

Time Division Multiple Access (TDMA) is where each earth station sends signals in bursts, interleaved with one another, at the same frequency. The number of burst time-slots in a given period of time can vary to enable the data rates to be varied, which is technically easier to achieve with TDMA than with FDMA. The traditional use of satellite is for long-haul inter-continental circuits; these commonly use TDMA. The way that these operate is that several earth stations in different countries all transmit bursts at a bit-rate of, say, 120 Mbit/s to the satellite which, when synchronised properly, occupy the entire power and bandwidth of a 72 MHz transponder. The earth stations can vary the lengths of their bursts according to how much traffic each has to send and in this way the transponder capacity is dynamically shared. TDMA systems tend to make very efficient use of the transponder but the earth stations tend to be large and expensive since each earth station has to receive the entire 120 Mbit/s multiplex even though they may only require a tiny fraction of it. Each earth station transmits for only a fraction of the time, in regular bursts, interleaved with each other. Earth stations monitor their own bursts on the downlink and uses this to accurately measure the round-trip time and hence to finely adjust the burst timing synchronisation. For example, an earth station with just one 64 kbit/s voice circuit to send will transmit for just 64/120,000 of the time, in one 'timeslot' (in practice a timeslot is slightly longer than this to accommodate overheads like synchronisation, error-coding and filtering). If the requirement for circuits goes up for one

particular earth station and down for another, the number of timeslots can be re-allocated by a control channel.

MF-TDMA

In order to achieve the advantage of TDMA and yet avoid the necessity for large and expensive earth stations, more modern systems use access schemes comprising a combination of FDMA and TDMA known as Multiple Frequency TDMA (MF-TDMA) where the multiplexes are smaller.

Some commercial VSAT systems use this form of multiple access on transparent satellites, for example Viasat [2-24] and Gilat SkyStar [2-25].

CDMA

The final type of multiple access described here is code-division multiple access. This has limited use on communications satellites because the number of simultaneous users is typically lower than when TDMA is used. The disadvantages with CDMA are (a) that the cross-correlation between codes produces mutual interference between users and (b) a large overhead is necessary for uplink power control so that all users are at equal power at the satellite receiver. However, CDMA has the attributes of allowing soft handover between satellites and being highly resistant to jamming. For these reasons it is used by Globalstar for voice and data and by military systems like Skynet.

2.4.8 Demand-assigned multiple access (DAMA)

With any of the above access sharing methods, the capacity may be reserved, which means it is available all the time, or it can be made available upon request. If capacity is made available upon request, the system is known as Demand Assigned Multiple Access (DAMA). Quite commonly a combination of reserved and demand-assigned is used and the balance between the two is adjusted according to traffic load.

A DAMA system requires a scheduler to allocate access, such a scheduler resides at a hub station in the case of a transparent satellite system, or on the satellite in an OBP system. Where DAMA is used, the delay between the request being made for resource and it actually being allocated will affect the delay experienced by user traffic and putting the scheduler on the satellite halves that delay.

The issue of Quality of Service (QoS) parameters such as delay and associated transponder efficiency with DAMA are the subjects of much current research. Two examples are:

- the University of York has performed research on the effect of bursty traffic on DAMA schemes and invented a new access scheme called PB-DAMA (piggy-back demand assigned multiple access) which piggy-backs information on burst size and queue length parameters onto the transmission so that these can be used in the algorithm for resource assignment. This mitigates the problem of infinite buffer requirements in the DAMA terminal when the offered traffic is self-similar. The university published a paper on this at an IEE seminar [2-26],
- Purdue university have explored packing algorithms for packets into MF-TDMA schemes to provide assured QoS and optimised this for maximum resource allocation efficiency [2-27].

Since DAMA allocates bandwidth on satellite links, it inter-works with the OSS especially in the areas of monitoring QoS parameters and prioritisation of traffic.

2.4.9 Reliable broadcast and multicast

Satellites are suited to broadcast and multicast services because of their high vantage points. Reliability of these services is a current research topic, brought about by high latency combined with the satellite link sometimes being one-way. Most of the research is concerned with data compression and / or techniques to enhance the performance of TCP when subjected to the latency of a satellite link. Such enhancements allow TCP-based applications like FTP and HTTP to be run with a user experience similar to that on terrestrial networks.

A comparison of reliable multicast protocols is presented in a paper by Levine et al [2-23] who has ranked them on the basis of their throughput. It is a good overview of the problems with reliable multicast, notably the implosion of receiver-initiated protocols such as the negative acknowledgement (NAK). They argue that NAKs can be effectively managed without infinite queues if the receivers are organised into a tree structure.

A reliable multicast transport protocol (RMTP) is proposed by Bell Laboratories [2-28] where a group of receivers in each region are contacted through a designated receiver. The designated receiver then collects the ACKs from end use receivers in their local region and takes care of re-transmissions in that area. RMTP is designed to be able to overlay existing multicast schemes such as Protocol Independent Multicast (PIM). It seems more suited to terrestrial networks that have several nodes with several spanning trees, rather than to satellite networks which will typically have one big spanning tree. For further information, an excellent description of RMTP is given by Sabnani et al [2-29].

A good description of the problem of reliable multicast by satellite is given by Gory Fairhurst et al [2-31]. The paper looks at star based, tree based and ring based organisations. It also contains a taxonomy of the various multicast protocols (RMTP etc) giving their reliability mechanism and commenting on their reliability and congestion control. Security is addressed briefly. There are lots of references to other multicast papers.

The EU funded ICEBERGS project has reported on work on QoS for real-time multicast over satellite and is described in a paper by Bianchi et al [2-32]. The paper discusses protocols for multicast over the EuroSkyWay on-board processing (OBP) satellite system, the protocols include Session Initiation Protocol (SIP) and Resource Reservation Protocol (RSVP). The reference contains a useful description of SIP that is defined by RFC 3261 [2-36]. The paper proposes a new admission control protocol for multicast that originated within SUITED (the predecessor to ICEBERGS) called GRIP (Gauge and Gate Reservation with Independent Probing) with multicast applications in mind. The paper offers nothing in the way of results, presumably because EuroSkyWay is not launched yet (latest estimate is end of 04 for first of two satellites).

Asynchronous Layered Coding Protocol instantiation is described by IETF RFC3450 [2-30] offers massively scalable solutions to the problems of reliable IP multicast, providing:

‘reliable asynchronous delivery of content to an unlimited number of concurrent receivers from a single sender’.

This is particularly suited to satellite systems and for content ‘push’ systems such as satellite broadcast. It works by inserting session description announcements that includes the sizes of objects to be sent, or by inserting residual times in the packet headers which can be used by the receivers to determine whether they have enough packets to reconstruct the objects. It is interesting that the list of authors on this RFC include M Luby from Digital Fountain (see below).

Another interesting approach to one-way reliable multicast is a product called Digital Fountain described in a white paper [2-34]. This patented protocol organises data into frames where each frame contains data about what is contained in all other frames (meta-data). Provided that delay is not critical, reliability of delivery is assured, since the receiver performs solutions to simultaneous equations to recover the user data and there will always eventually be enough information for the receivers to do this.

2.4.10 Doppler and network synchronisation

Doppler shift to the signal frequency is caused by the satellite changing range, when the satellite moves in the sky. A GEO satellite will generally move slightly on a daily cycle and therefore have a low Doppler shift, whereas LEO and MEO satellites move quickly across the sky and have high shifts. The RF carrier frequency shifts and so does the data rate. Unless properly managed, the shifts in data rate can cause data loss when connecting the satellite link to synchronised terrestrial networks.

A 4 degree inclination on a GEO satellite causes a 24-hour period sinusoidal Doppler shift of about +/- 200 parts per million, depending on relative positions of the satellite and earth stations. Since the data rate shifts on a daily cycle, it is generally necessary to buffer the data in the satellite demodulator before it is transferred to the receiving network. The reason is that the receiving network will most likely have a master clock and is expecting data at a fixed rate. Most satellite demodulators are fitted with such a ‘Doppler buffer’, which is filled from the satellite and emptied using the receiving network clock.

Figure 2-3 shows one such arrangement.

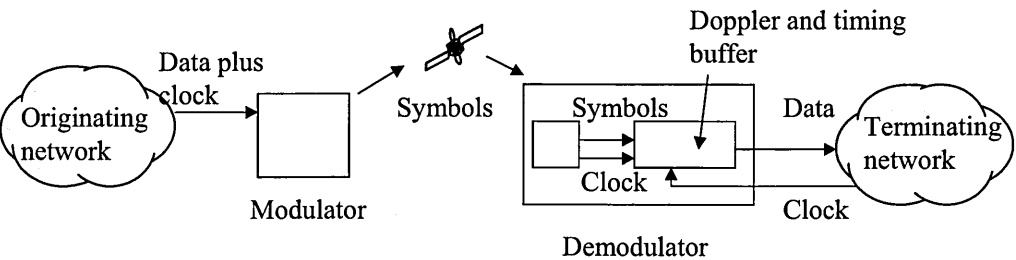


Figure 2-3 A typical arrangement for clocking and Doppler buffering on a satellite link

In figure 2-3, the modulator transmits symbols towards the satellite that are synchronised with the originating network clock. In the demodulator, the symbols and the clock recovered from the symbols, which are affected by the Doppler shift, are fed into the Doppler and timing buffer. The data is clocked out of the buffer by the terminating network clock. In this way, the buffer smooths out the variations in timing caused by Doppler shift and it compensates somewhat for slight differences in the clocks in the two terrestrial networks. Such buffering adds a small amount of delay.

Processing satellites do not translate Doppler from uplink to downlink and so the Doppler shift from these is less.

2.5 The changing use of satellite

2.5.1 Changes in the marketplace

The use of satellite is changing, yet they have to remain competitive to survive, which is one key reason for performing this research on satellite OSS. Figure 2-4 shows the various places that satellites can be used in a communications path, in relation to different types of networks. Also included on the figure are terrestrial technologies that compete with satellite.

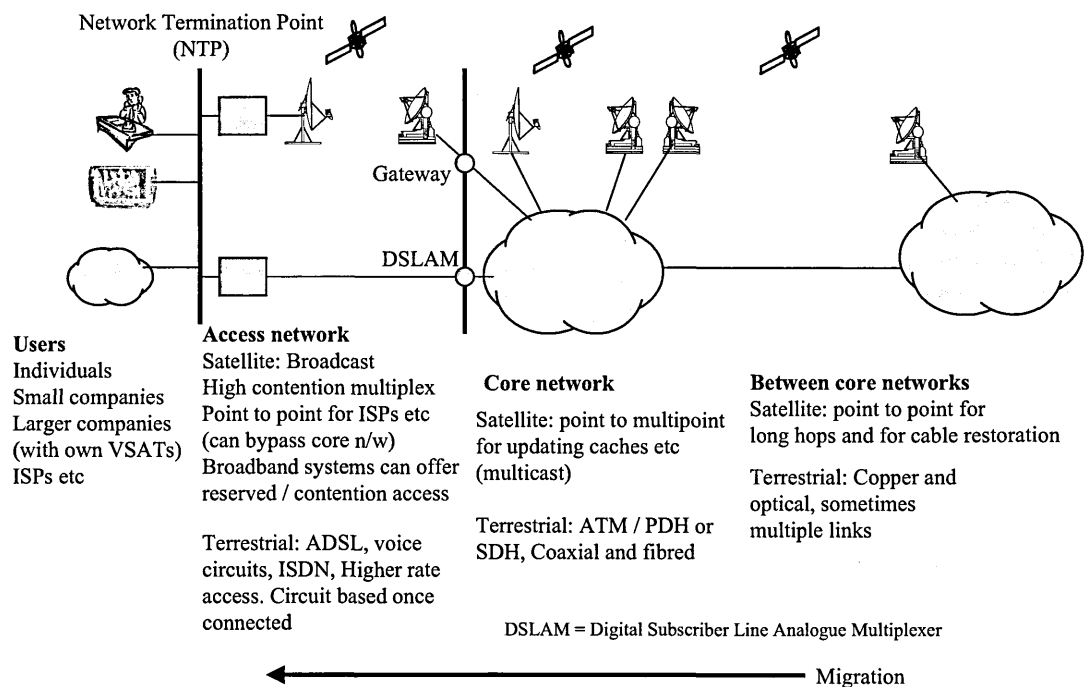


Figure 2-4 Where satellites are used

The traditional use for satellite is depicted on the right-hand side of figure 2-4, between core networks using large earth stations, typically with antennas that are 10 – 20m diameter. The applications here are trunk telephony (voice) and switched and un-switched data services, sometimes as the primary communications between countries and sometimes as back-up for cable systems. The data rates are commonly over 100 Mbit/s per beam and consist of many

circuits multiplexed together. The competitive technology here is under-sea fibre and copper systems and, where there are fibres installed (such as across the North Atlantic ocean), the cost of transmission is reduced significantly and satellites are generally no longer commercially viable². The decline of use of satellite for inter-continental communications is being driven by the installation of fibre-optic cable to more countries. But those trunk satellite routes that remain are high-value routes: if satellite is the only option, then users will pay and hence Service Providers charge high rates for these links, whatever the market can stand.

Moving to the left of figure 2-4, the next section describes distribution of data to the edges of core networks where there are large memory caches. This is a recent application for satellite, for non real-time data, announced by the British National Space Centre in 2003[2-35]. The satellite system off-loads non real-time traffic from the core network and, because the transmissions can be repeated from a carousel, use can be made of lower cost satellite capacity³.

Moving further to the left, the access network is encountered, where individual users have small satellite earth stations installed on buildings (e.g. offices and homes). It is here that the VSAT is used; depending upon whether it is one-way or two-way, the size of a typical VSAT is from 60cm up to 3.4m diameter.

The use of satellite is moving to the left of figure 2-4, a migration as time goes by. This is significant, because it means that satellite usage is migrating from a few users with large earth stations and high-value routes, towards large numbers of users with small earth stations and low-value routes.

Further information on the state of the satellite market, including revenue predictions, can be obtained from market survey reports by publishers such as Bear and Stearns [2-11].

So the satellite industry is being squeezed from its traditional markets due to the growth of long-distance fibre and the challenge is to reduce the cost-base in the content distribution and access network markets so that they grow sufficiently quickly to balance the loss.

As stated in chapter 1, from the OSS perspective this change of use of satellite and required reduction in cost-base is driving towards more integration and automation. The problem is that OSS for satellite systems is currently largely fragmented, system specific and not at all integrated or automated. This theme is picked up and addressed in chapter 3.

2.5.2 Satellite operator response to the changes in the marketplace

Satellite operators such as Intelsat, Eutelsat and Inmarsat have had to re-organise themselves to cope with de-regulation and the change in market away from core interconnect towards the access network. Up until the 1990s they were co-operatives, which were government treaty organisations with countries as owner-signatories. In the UK, the government appointed

² There is one notable exception, which is the provision of asymmetric or one-way data services for, say, Internet Service Provider (ISP) backbone connections. Satellites are very well suited to this kind of service and can sometimes compete effectively with fibre.

³ Higher satellite frequency bands (e.g. V-band at 47 – 50 GHz and Ka-band at 20 / 30 GHz) tend to be more unreliable due to the effects of weather and so capacity may be offered at lower cost.

British Telecommunications plc as manager of signatory affairs for all three of these co-operatives.

Intelsat is a global operator that has specialised in core interconnect traffic, using high bandwidth satellites mainly in the C and Ku bands for inter-continental trunk traffic since the 1960s. Eutelsat are a European regional satellite operator specialising mainly in Ku band TV transmission although they also offer limited Ku-band mobile services. Inmarsat are a global satellite operator that has specialised mainly in maritime and aeronautical services (including emergency) using L-band to the terminals. All three of these operators have expanded their operations to cover land-based fixed and mobile services and are seeking to move up the value chain by providing a greater range of services over and above basic transmission. Value chains for satellite communications are discussed further in chapter 3.

Figure 2-5 illustrates an example of how a co-operative satellite operator worked in terms of flow of money when a satellite is purchased and launched.

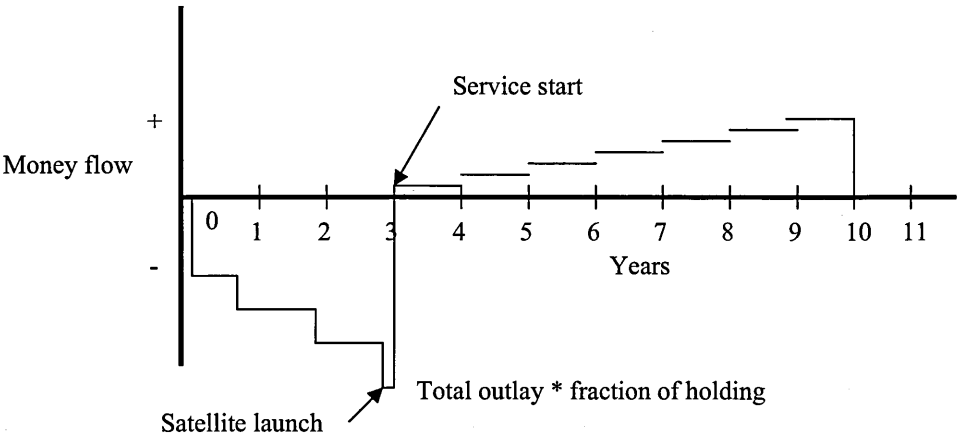


Figure 2-5 Money flow when a co-operative launches a satellite

The satellite is produced by a space manufacturing company and is usually paid for in stages according to milestones. This is shown by the stepped negative-going outlay. Each signatory to the co-operative pays a fraction of these payments according to how much of that satellite they will wish to use. For example, the UK may commit to using 10% of a new satellite, in which case the UK pays 10% of the price. This coupling of the investment to usage is a characteristic of the co-operative method and was one key factor that led to their changing, since it limits flexibility.

After launch, the signatories receive typically 20% of their investment each year for 10 years, regardless of satellite lifetime, which is a very lucrative return. After 10 years, no more money is received, the revenue stays with the co-operative. Therefore the co-operative carries the risk of satellites failing in early life. Typical lifetimes are 12 – 15 years.

The co-operative arrangements lasted for 30 years until the late 1990s and solved problems such as funding the purchase of new satellites (which was done entirely with equity) and obtaining landing rights for the coverage. However, it also had the following drawbacks:

- that the same price had to be charged for capacity regardless of geographical region,
- that signatories utilised the capacity according to their ownership fraction,
- that approval from all signatories was required for all spend.

The first drawback meant that prices could not be altered to regulate demand in the different parts of the world, the other two drawbacks meant lack of flexibility and ability to change quickly as the market changed. The response was that the co-operatives have now all privatised (the last one in 2000) and the signatories became shareholders. The satellite operators now have conventional shareholders and fund purchases of new satellites through debt; this has opened the way for many smaller satellite operators to start up, so that service providers now have a wider choice of sources for satellite capacity.

From the OSS viewpoint, having more satellite operators increases the importance of an integrated management approach based on standards.

2.6 Some example systems

The final section in this chapter gives some information on some example systems, some operating and some not. Table 2-2 contains further information on the various satellite fleets mentioned so far in this chapter, plus some others. Information is mostly taken from Janes Space Directory [2-37]. Janes also has information on the many other smaller satellite operators, mainly run by individual countries, e.g. Apstar, Hispasat, Aussat, France Telecom and Italsat.

Fleet name	Satellite type	Orbit type	Number in fleet	Coverage	Operating band	Status at end of 2004
PanAmSat	Transparent, 36 and 72MHz transponders	GEO	21	Global	C and Ku band to fixed terminals	Operational
Intelsat	Transparent, 72MHz transponders	GEO	21	Global, by region	C and Ku band to fixed terminals	Operational
New Skies	Transparent, 72MHz transponders	GEO	5	Global, by region	C and Ku band to fixed terminals	Operational
Eutelsat	Transparent, 36 and 72MHz transponders	GEO	8	Mainly Europe	Ku band to fixed earth stations (mainly TV)	Operational
SES Astra	Transparent, 36MHz transponders	GEO	12	Europe	Ku band to fixed earth stations (TV). Two satellites have Ka band transponders	Operational
GE Americom (owned by SES)	Transparent, 36MHz transponders	GEO	19	North America	Ku band to fixed earth stations (TV).	Operational
Inmarsat I – III	Transparent 100MHz transponders ⁴	GEO	7	Ocean regions	L-band to mobile terminals	Operational
Inmarsat IV	Transparent, digitally filtered	GEO	3 planned	Global, by region	L-band to mobile terminals	Launch during 2005
Iridium	Processing	LEO	66 plus 6 spare	Global	S-band to handsets	Launched, main customer is US DoD
Teledesic	Processing	LEO	288 planned	Global	Ka band to fixed terminals	Abandoned, not launched
ICO	Processing	MEO	12 planned	Global	S band to handsets and mobile terminals	Two launched for spectrum planning
Spaceway	Processing	GEO	9 planned, cut to 1.	Global, by region	Ka band to fixed terminals	Changed to provide HDTV services
Globalstar	Processing	LEO	48 plus 4 spare	Global	S band to handsets and mobile terminals	Launched but bankrupt. Owner not known
EuroSkyWay	Processing	GEO	5	Europe and North Africa	Ka band to fixed and mobile terminals	Not yet launched, date unknown.

Table 2-2 Some satellite systems

⁴ To cover the MSS band 1.5 to 1.6 GHz

2.7 References for chapter 2

- [2-1] G. Maral and M. Bousquet. *Satellite Communications Systems*. Second edition. Wiley, 1993. ISBN 0-471-93582-4.
- [2-2] *EADS Astrium-built Intelsat 10-02 spacecraft shipped to Baikonur*. EADS press release, May 18th 2004. See: http://www.space.eads.net/web1/press/press_release.asp?langue=en&id_tree=320&id_tree_nav=77. Link checked 28th December 2004.
- [2-3] *Integrated SDH L-band Modulator board. Newtec Azimuth series*. Data sheet available from <http://202.83.98.162/datasheet/ntc.7055.ds.pdf>. Link checked 28th December 2004.
- [2-4] Brochure on Astra 1C. Available from http://www.ses-astra.com/press-info/documentation/brochures/pdf/0_1C_Footprint_FactSheet.pdf. Link checked 28th December 2004.
- [2-5] John Puetz. *Integration of Satellite and Terrestrial*. Satellite Communications Magazine, November 1999. Page 27.
- [2-6] RJ Watson, A Page and PA Watson. *Time-series of attenuation on EHF and SHF fixed satellite links derived from meteorological and forecast data*. IEE Seminar on Personal Broadband Satellite, July 22nd 2002.
- [2-7] Draft new Recommendation ITU 3/90-E. *Prediction method of fade dynamics on Earth-space paths*, 31st May 2002.
- [2-8] COST is a European forum for co-operative scientific research. For further information on COST action 280 see <http://www.cost280.rl.ac.uk/>. Link checked 23 February 2004.
- [2-9] *Propagation Data and Prediction Methods required for the Design of Earth-Space Telecommunications Systems*. Recommendation ITU-R P.618-7, ITU-R Propagation Series.
- [2-10] Stephen Cable. *Network Bandwidth on Demand for Multimedia and Internet Applications*. Satellite Communications magazine, May 1999. Pages 48 – 53, figure 1.
- [2-11] Bear and Stearns. *Satellite Communications*; Equity report: Fixed Satellite Service, September 2001. Page 123.
- [2-12] International Telecommunications Union. *Radio Regulations part 1E 'Articles'*, Geneva 1998.

- [2-13] European Technical Standards Institute. *Framing structure, channel coding and modulation for 11/12 GHz satellite services*. Publication EN300 421. Digital Video Broadcasting (DVB); August 1997.
- [2-14] F Petz, J Ventura-Traveset, I Stojkovic, J Alonso. *Technology review of present and future personal communication satellite payloads*. European Space Agency. IEEE Microwave Systems Conference, 1995. Pages 135 – 144.
- [2-15] Hamidi, Wisewell, Cherrette, Saunders, Ho. *TRW's Broadband Communications Payloads and C and KU Frequency Bands*. IEEE Aerospace Conference Proceedings 9 – 16th March 2002. Page 3-1253.
- [2-16] KDD R & D Labs. *Analysis and Dimensioning of Ka-band System Capacity*. Study for Intelsat (INTEL 2090), final report. Undated.
- [2-17] R Subbarayan. *Intelsat Broadband Access Services*. Global Telecommunications Meeting 2002, Washington DC May 2002, Special Interest Session.
- [2-18] M Spagnulo, G Matarazzo, F Rispoli. *Market and Business perspectives for Euroskyway, the European Broadband Satellite Network*. Alenia Aerospazio, 1999.
- [2-19] Hughes Network Systems. *Spaceway overview*. See <http://www.hns.com/HNS/Rooms/DisplayPages/LayoutInitial?Container=com.webridge.entity.Entity%5B0ID%5B247EE93EA910FC4882B3105E9AD73791%5D%5D>. Link updated on 21st December 2004
- [2-20] European Space Agency Press Release. No 7- 1998. *Skyplex, The First Multimedia Processor in the Sky*. 26th February 1998. Available from http://www.esa.int/export/esaCP/Pr_7_1998_p_EN.html Link checked 21st December 2004
- [2-21] Hugh Arif. *Internet 3 – Internet Protocol in Space*. Presentation to Satellites in IP and Multimedia Workshop organised by ITU in Geneva. December 10th 2002.
- [2-22] G. Maral and M. Bousquet. *Satellite Communications Systems*. Second edition. Wiley, 1993. ISBN 0-471-93582-4. Page 7
- [2-23] Levine, Garcia-Luna-Aceves. *A comparison of reliable multicast protocols*. Multimedia Systems 6, 1998. Pages 334 – 348.
- [2-24] Viasat Products and Technologies. See <http://www.viasat.com/vsat/linkway/> Link checked 21st December 2004.
- [2-25] Gilat Skystar Technical Specifications. See www.gilat.com/Products_SkyStarAdvantage_TechnicalSpecs.asp?Sbj=345 Link checked 21st December 2004.

- [2-26] Paul Mitchell, Tim Tozer and David Grace, University of York. *Access Techniques for Broadband Satellite*. Personal Broadband and Satellite Seminar. IEE, Tuesday 22nd January 2002.
- [2-27] Jung Ming Park. *Efficient Resource Allocation for QoS Channels in MF-TDMA Satellite Systems*. IEEE 0-7803-6512-6. 2000.
- [2-28] Bell Laboratories. *RMTP: A Reliable Multicast Transport Protocol*. Available from <http://www.bell-labs.com/project/rmtp> Link checked 21st December 2004.
- [2-29] Paul, Sabnani, Lin, Bhattacharyya. *Reliable Multicast Transport Protocol (RMTP)*. IEEE Journal on Selected Areas in Communications. Vol 15, No 3 April 1997. Pages 407 – 421.
- [2-30] Luby, Gemmell, Vicisano, Rizzo, Crowcroft. *Asynchronous Layered Protocol Instantiation*. IETF Networking Group RFC 3450, December 2002.
- [2-31] Martin Koyabe and Godred Fairhurst. *Reliable Multicast via Satellite – A Comparison Survey and Taxonomy*. Electronics Research Group, University of Aberdeen. Undated.
- [2-32] Bianchi, Blefari, Marziale et al. *QoS for Real-Time Multicast over Satellites*. ICEBERGS project report to the European Commission. Undated.
- [2-33] Conforto et al. *Ubiquitous Internet in an Integrated Satellite-Terrestrial Environment: The SUITED solution*. IEEE Communications Magazine, January 2002. Pages 98 – 107.
- [2-34] Digital Fountain White Paper. *Next Generation Data Transfer: The meta-content Revolution*. 2002. Available from <http://www.digitalfountain.com>. Link checked 21st December 2004.
- [2-35] D McGovern and A Fidler. *B3*. Proceedings of Working with the Telecommunications and Navigation Advisory Board. A joint IEE / BNSC event, 22nd October 2003.
- [2-36] *SIP: Session Initiation Protocol*. RFC 3261, Network Working Group, IETF, June 2002.
- [2-37] Jane's Space Directory 2001-2002, page 336

3 Overview of Operational Support Systems (OSS) and their relationship with satellite services

3.1 Introduction

This chapter aims to give an overview of OSS, what it is, what it does and how it is currently applied to satellite systems. Section 3.2 gives an overview of OSS and section 3.3 describes its key business drivers. Section 3.4 is a review of OSS architectures and technology, including protocols and distributed and web-based architectures. Section 3.5 gives some examples of commercial off-the-shelf (COTS) OSS software. Section 3.6 is a review of OSS industry models, which is largely the work of industry forums and standards bodies. The output from these sections is a choice of business framework (the eTOM) on which to perform an analysis of what is different between satellite and terrestrial OSS.

Section 3.7 gives a review of OSS research activities, which focusses onto satellite-specific OSS towards the end of the section, followed by section 3.8 which describes some examples of practical implementations of satellite OSS.

Section 3.9 begins a new thread by developing a value chain, which is applied to a practical example, the provision of satellite TV by Sky. The essential arguments from this section are that there are definite boundaries for management responsibility and that one actor typically needs to reach through another actor in order to fulfill his management role.

Finally, section 3.10 describes the analysis and identification of differences between satellite and terrestrial OSS processes.

3.2 What is OSS

OSS are defined by the Telemanagement Forum (TMF) [3-1] as:

‘systems that an organisation must put in place to provide fulfillment, assurance and billing (FAB) of the services that it sells’.

OSS are a special type of information system addressing the needs of communication industries, network operators and service providers in providing FAB for a wide range of network based products and services to their many customers. As such, OSS reflect the multi-faceted demands of the many types of businesses they support.

The field of OSS is reasonably new but it is developing rapidly. In the late 1970s the scope was limited to the PTTs and their bespoke solutions to address their specific business goals. An OSS community and then an industry developed via the rapid growth of the mobile operators such as Cellnet and Vodafone. The emergence of OSS application supply businesses was further stimulated by de-regulation and competitive carrier investment through the boom in the 1990s. The OSS industry has consolidated post-boom through the re-discovery of traditional business values (such as customer care) and the advent of next generation networks (NGN).

Some of the key trends in the short history of OSS are:

- OSS have evolved from a position where they supported islands of automation only, to one where end-to-end integration of business processes and automation is the target,
- Increasingly comprehensive industry models and architectures have evolved and now provide a common foundation for company architectures,
- Many OSS are now built from a base of common industry supplied commercial off-the-shelf (COTS) applications, some generic (sometimes called industry horizontal) applications such as Customer Relationship Management (CRM) and some quite communications sector specific (vertical applications), such as network management,
- There is a strengthening value chain of OSS supply; Telecommunications equipment vendors, Infrastructure and Application vendors, Systems integrators and the IT and CIO functions in operators and service providers,
- There is a growing base of industry OSS standards being developed through the efforts of OSS product vendors, industry forums such as the Telemanagement Forum (TMF) and standards bodies such as the ITU-T, ETSI, IETF and IEEE.

A white paper by Compaq on OSS solutions for IP [3-2] concurs with the above trends and concludes:

‘In view of these trends, it seems likely that there will be massive replacement programs in almost all areas of IP Network and Service Management in the coming years.’

Communications services are now delivered via a complex value chain of Network Operators, Service Providers and mobile operators. The OSS needs of these actors are distinct and specialised and there is a growing need to support the interaction between these operators and service providers via business-to-business (B2B) trading capability.

Large operators such as telcos tend to require OSS that support high levels of automation and a broad range of products and platforms. They also have the longest history and the deepest legacy systems base. Small operators such as competitive carriers or Internet Service Providers (ISPs) tend to rely more on skilled people with lower levels of automation, they have systems and networks more focussed to the specific areas they compete in and they tend to use more COTS OSS products.

As Next Generation Networks (NGN) are laid down, the successful communications businesses will be those that can align to and anticipate their customers’ needs and provide the integrated services they require at world best cost. OSS will be one of the critical differentiating elements in enabling world-class customer service by providing the necessary automation. NGN, according to ETSI [3-3] is:

‘a concept for defining and deploying networks, which, due to their formal separation into different layers and planes and use of open interfaces, offers service providers and operators a platform which can evolve in a step-by-step manner to create, deploy and manage innovative services’

From the point of view of OSS, two essential characteristics of NGN are that it provides capability to create and manage many kinds of services and has functional entities that can be distributed over the infrastructure via open interfaces.

NGN are seen increasingly by network operators as converged networks, where mobile and fixed services can run over the same infrastructure.

3.3 OSS key business drivers

The key business drivers for service providers' OSS are common to all companies, i.e. to support the primary value chains around the FAB of key products. OSS should also support the effective launch of whatever new products are required to maintain competitiveness in line with business strategy.

OSS business cases are typically based upon a combination of three key drivers:

- **Minimise Operations Cost** – OSS play a key role in minimising the costs of providing and maintaining service. Lower operations costs from higher levels of automation in FAB are often also associated with better customer oriented measures on Service Level Agreements (SLAs) e.g. provide repair time and accurate billing information. These in turn can support improved customer satisfaction. High levels of automation are necessary for mature products and 'flow-through automation' of work within service provider and between service providers via B2B automation where they jointly deliver service to customers. OSS that reduce the operational costs will allow the capital expenditure to be used more effectively,
- **Maximise Profitable Revenue** - OSS are necessary to support the launch of new products and services. In this case the OSS investment has to be taken into account in the business case for a new product. However OSS are increasingly required to exhibit the kinds of flexibility and extensibility to support the rapid launch or trial of new products or new features on established products. Increasingly OSS will be part of the service itself and delivered via Web or other devices supporting electronic commerce. Self service is becoming the norm,
- **Maximise the Return on Capital Investment** – the communications industry spends heavily on network technology year on year – the costs of OSS tend to be a small percentage of that capital investment. In an increasingly competitive environment OSS help operators to ensure that they can run their networks to deliver optimal network utilisation leading ultimately to lowest unit costs. The approach used to deliver optimal utilisation may vary. For example, a strategy to grow market share would be 'build then sell' in key target customer segments and geographies until the target market share was achieved. A market defence strategy typical in mature markets may lead to lower stocking levels, and an approach of 'sell then just-in-time build' leading to lower unit costs.

The focus and approach on OSS will develop as the market develops, this is illustrated in the S-curve in figure 3-1.

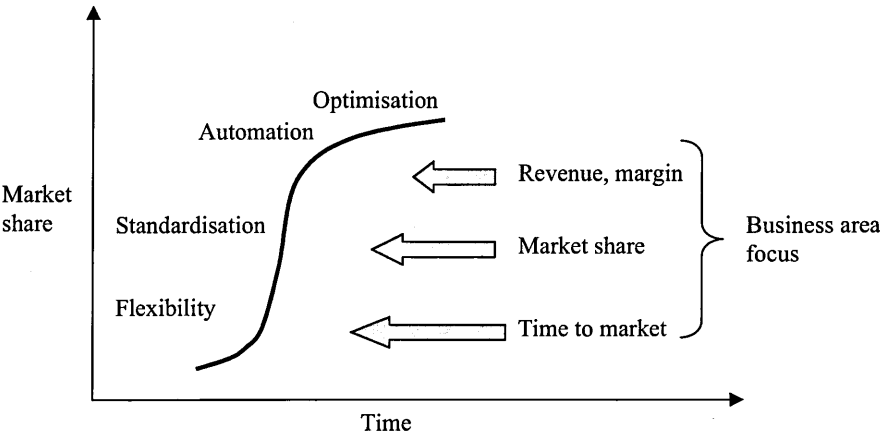


Figure 3-1 The OSS S-curve

To put a satellite perspective to the time-scale on figure 3-1 with an example system under development, Inmarsat are paying Astrium £700m for three processing satellites to form their IV series that will be launched in 2005. As this purchase is funded by debt, then early year revenues are critical and, if these are not on target, the payback period can easily be beyond the life of the satellites (typically 13 years). Replacement satellites may then have to be launched before the operation becomes profitable, especially if the satellite system supports only a restricted set of services (e.g. mobile to handsets). With LEO systems the early year revenues are extremely important, because much or all of the constellation has to be present before service can start and the lifetime of the satellites is less (typically 7 - 10 years).

Therefore, because of the large up-front investment with satellite systems, a very important attribute of OSS at the low end of the S-curve in figure 3-1 is to enable fast customer sign-up at the beginning of service and to increase market share as quickly as possible. One reason for Iridium and Globalstar going into receivership was that customers were not signed up quickly and they were in competition with the fast-expanding GSM operators. The mid-point of the S-curve for these examples will be about 2 years after launch, with the optimisation phase at about 4 years from launch.

So in general, OSS should deliver business-oriented solutions matched to the priorities of the business it supports. Communications and network related products often also follow the lifecycle described in the S-curve in figure 3-1. In the early life stages flexibility and speed are often most important as it can be important to experiment with features and to assess which will succeed in the marketplace. The biggest area and impact on OSS is for communications and network-related products and network technologies that manage to reach high volume. For rapidly growing products, OSS is key to enable volumes and revenue growth with manageable

OPEX around standardised fulfillment and assurance processes and increasingly standardised products; in later lifecycle OSS supports much tighter cost control, both OPEX and CAPEX, and incremental revenue opportunities.

3.4 Review of OSS architectures and technology

3.4.1 The management pyramid

The starting point is the rather well-known Telecommunications Management Network (TMN) management pyramid, shown in figure 3-2. All known OSS stacks are based on this model. It is derived from the description of management layers in ITU-T recommendation M3010 [3-4].

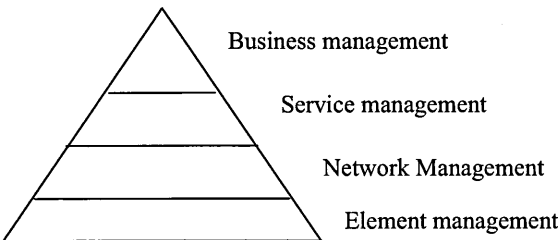


Figure 3-2 Management pyramid

Starting from the bottom of figure 3-2, the element manager is the lowest of the managers. It interacts with the network elements, which are the actual boxes such as routers, switches and satellite modems. This layer can be cascaded, i.e. one element manager can be connected to another in a form of master-slave arrangement, for example as shown in figure 3-3.

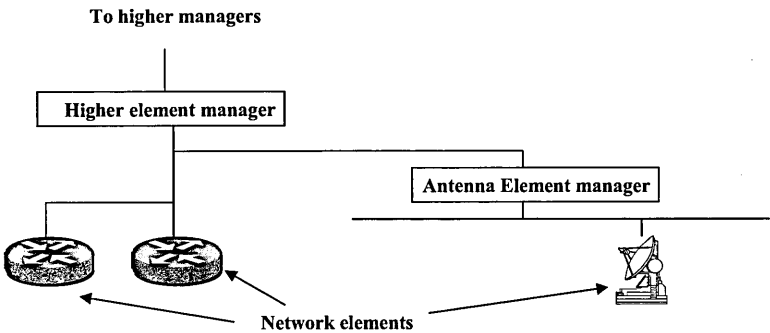


Figure 3-3 Cascaded element managers

In figure 3-3, the antenna element manager is dedicated to the antenna, but has a gateway to a higher element manager. To the higher element manager, the Antenna system could be one icon showing summary alarm status, configuration etc, while the detailed configuration settings for the antenna would be performed from the dedicated manager.

Cascading is quite common practice with satellite element managers, where the management system is purchased along with the satellite system and it has a gateway to a higher element manager that may be responsible for whole portions of the network. Some examples of commercial systems using this technique are given in section 3.7.

Moving up figure 3-2, the network management level is responsible for providing network resources to the service management layer. It keeps track of the network topology (which is not a trivial task) and has rules which re-route traffic around faulty network segments. It correlates alarms from the element layer and performs fault analysis. It also keeps statistics and logs of network usage and interacts with the service management layer on performance.

The service management layer interacts with the service provider and with other management stacks, maintaining service statistical data (such as QoS). It co-ordinates and interacts between services and will know which services are affected by any network faults. It is at this layer that the OSS stack will communicate with other stacks if the service to be provided requires resources from more than one network. This important feature is described in more detail in section 3.8.

The business layer supports trend and policy information for decision making on investments and usage of network resources. It also supports the management of maintenance budget and manpower. A customer care package, which will have processes such as fault reporting and order handling will interface to the service management layer.

3.4.2 OSS protocols and the role of managers and agents

Simple Network Management Protocol (SNMP) is the most widely used standard network management protocol and it has emerged from the networking community to fulfill the need to manage network elements as described in the previous section. SNMP is used at the lowest (element) management layer in figure 3-2.

If compatible with SNMP, the element will have an SNMP agent for communicating with an element manager. SNMP was developed by the IETF (Internet Engineering Task Force) during the 1980s. The latest version, SNMPv3 addresses one of the most significant weaknesses of earlier versions, in the area of security. It is described in a set of RFC (Request for Comments) documents 2271 – 2275.

SNMP is mandated to use UDP/IP (User Datagram Protocol), which is not a reliable transport protocol. SNMP is therefore best suited to management of elements over local area networks (LANs). In order to get around this for wider use, some element management solutions encapsulate SNMP into proprietary protocols to increase reliability; an example is the PanaVue Management platform for the Promina ATM switches [3-5].

Whereas SNMP is suitable for smaller networks, for more general cases the OSI model provides a more powerful infrastructure by adopting an object-oriented approach. In this model, elements are represented through the use of managed object abstractions that can be handled by agent applications and accessed by manager applications. Communications

between these agents and managers is achieved through the OSI Common Management Information Service / Protocol (CMIS / P) [3-6], [3-7]. CMIS provides the OSI management services to the management applications, and CMIP provides the information exchange capability to support CMIS.

Therefore SNMP and CMIP fulfill basically the same function over different types of networks; they are both protocols that allow managers to communicate with agents. However CMIP is also applicable to peer communications at the service management layer whereas SNMP is used almost exclusively for element management. CMIP runs over TCP, which increases its reliability over wide-area networks.

The element will have a set of parameters that can be managed or monitored, such as queue limits, packet counts, transmit power and alarms. This set of parameters is held in a data-base called a Management Information Base (MIB) that resides on the element manager. MIBs are written formally in Abstract Syntax Notation 1 (ASN.1) [3-8] template language using the Guidelines for the Definition of Managed Objects (GDMO)[3-9]. This object-oriented approach enables managed object classes to form an inheritance tree. MIBs for some common elements, such as Cisco routers, are freely available. Figure 3-4 shows the location of MIB and agent.

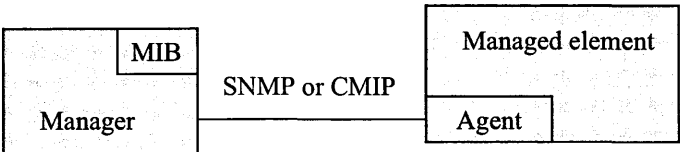


Figure 3-4 Location of MIB and agent

If the element is SNMP or CMIP compliant, the agent resides in the element to update the MIB. Figure 3-5 shows how, if the element is not SNMP or CMIP compliant, the element can still be managed by fitting an external agent, called a proxy agent.

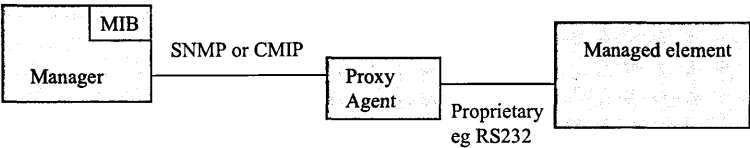


Figure 3-5 Use of a proxy agent

Scripts or processes in the proxy agent convert the SNMP or CMIP to the protocol that the element understands, for example RS 232 serial interface command-line or even, in the case of alarms, it can simply be pairs of relay contacts. Proxy agents are usually proprietary to the element for which they were designed and can be run on PC platforms.

The difference between SNMP and CMIP is explained in some detail by Dymek [3-10]. SNMP is, as its name suggests, much simpler than CMIP; according to the standards documentation, SNMP contains only five basic instructions, which are:

- Get Request (get a MIB variable),
 - GetNext Request (get the next variable in a table),
 - GetPrev (get previous variable) and GetBulk (get bulk values) is described by Breitgand for inclusion into SNMPv2 [3-11],
 - Get Response (answer to a Get, GetNext or Set request),
 - Set Request (change a MIB value),
- There is also the 'trap' message used by the element to report an alarm.

The key set of CMIS services are:

- Event-report, e.g. for unsolicited alarms,
- Get, i.e. data retrieval,
- Cancel Get,
- Set, i.e. parameter modification,
- Action, e.g. user defined action,
- Create, e.g. create instance of managed object,
- Delete, e.g. delete instance of managed object.

The advantages of using SNMP are:

- It is simple and widely implemented,
- It is light-weight requiring less memory and CPU cycles than CMIP.

The disadvantages are:

- It is constrained to use UDP and therefore not suited to wide-area networks,
- It does not scale well due to reliance on polling,
- It has a limited instruction set and is not suited to management above element layer.

SNMP, although simpler than CMIP, does not necessarily generate less management overhead traffic. SNMP operates mainly by polling whereas CMIP generates unsolicited events, which means that SNMP generates traffic even in the absence of faults or alarms. SNMP does not scale well with increasing network size because it uses polling and because most of the intelligence resides in the manager. Yet despite these limitations SNMP is popular because it is easy to use and is suited to small networks where bandwidth is not expensive. Since SNMP is both bandwidth-inefficient and unreliable, CMIS/P may be the better option for managing remote elements over a satellite link, but strangely there are no known commercial implementations.

However CMIP has been used in some test-beds and research projects over satellite. An example is the use of CMIP to remotely configure a satellite modem during a European Commission 3rd framework project called VANTAGE, the full account is published in the final report[3-12]. Of the range of requests available within CMIS, use was made of GET, ACTION, CREATE and DELETE of certain modem parameters such as Receive level, Go to transmit, Change coding scheme and Delete alarms.

3.4.3 Distributed architectures and associated software

SNMP allows exchange of parameters across networks, but as it is not object-based it does not lend itself to the possibility of distributed managers and agents. Since CMIS is object-based, it can be used on distributed management systems using a suitable architecture. A software infrastructure commonly called 'middleware', in the client / server distributed computing environment, provides consistent mechanisms for distributed software 'components' such as agents and managers, to communicate with each other across a network.

The middleware is thus responsible for organising or brokering the communications between distributed clients and servers. Above this, the application processes commonly call one another using remote procedure calls (RPC) that are analogous to functions calls, in that calling arguments are passes to the remote procedure and the caller waits for a response to be returned. By using RPC, the programmers of distributed applications avoid the details of interfacing with the network. An RPC can establish a server on a remote machine that can respond to queries and then retrieve information by calling a query.

According to Tang [3-13], the services provided by a middleware include:

- Naming service, to administer unique names for machines, users, files, access control groups, applications programs and to bind these with specific location addresses within the domain,
- Access services, to provide location transparency to communicating components. A client can gain access to a server by using the correct name without referring to the server's location in the network,
- File service, to provide a global distributed file system for storing and retrieving data,
- Security service, to authenticate users,
- Network time, to distribute accurate time across the network.

The field of middleware is moving very quickly and any snap-shot will quickly be out of date, but currently there appear to be two software middleware standards with significant support in the computing industry, that provide all of the above services. These are the Common Object Request Broker Architecture (CORBA) sponsored by the Object Management Group (OMG) and Java / Remote Method Invocation (RMI) from Sun Microsystems

CORBA uses Interface Definition Language (IDL), which gives the client and server the ability to specify their interfaces, after which the IDL compiler generates two interface 'stubs', one for the client and the other for the server. The CORBA IDL can generate interface stubs from other stubs with inheritance and can generate stubs in different languages. Further, if a client or server interface is not known at compilation time, CORBA provides a dynamic invocation interface (DII) that allows clients and servers to discover one another and to construct calls at run-time. Figure 3-6 illustrates the role of the IDL compiler in CORBA.

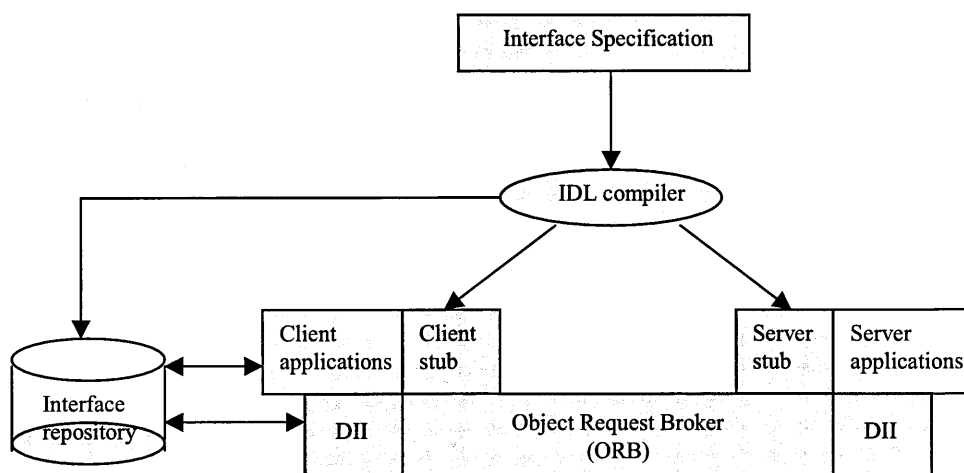


Figure 3-6 Role of IDL Compiler (shown with CORBA)

In a CORBA network management system, the manager is the client and the agent is the server. The DII allows managers to discover new agents and is a valuable feature when network topology is changing. Keeping track of the topology is one of the big problems with network management. Communication between the managers and agents is accomplished using mechanisms supported by the ORB. CORBA specifies a set of inter-ORB protocols, which allows the interaction among objects residing in ORBs from different vendors. The communication between and within ORBs is based on Internet Inter-ORB protocol (IIOP), which runs over TCP/IP and is therefore reliable and can be used over large distances.

Unlike SNMP, but in common with CMIP, the types of requests a manager may send are not restricted. In a CORBA-based management framework, MIB objects are defined as IDL interfaces and management functions are accomplished using invocation of object operations.

Through the work of the Joint InterDomain Management group (JIDM), which is working towards the co-existence of TMN and CORBA for Service Management, a CORBA agent can now be implemented as a Java applet. According to Djalalian, this permits the implementation of a graphical network management interface that is downloaded from an HTTP server and viewed via a WWW browser [3-14]. The local Java machine activates the applet, which acts as a CORBA client that communicates with the CORBA agents using IIOP. CORBA agents can also communicate with one another using IIOP in order to manage the whole network. This co-operation is based on specific network managed object interfaces that one CORBA agent can provide to others. In this way CORBA agents can provide management services to other agents and act as manager to agents of other management platforms.

An example of how far the integration of Java and CORBA has progressed is given in a paper by Barotto et al [3-15], who built a test-bed using SNMP, Java and CORBA to manage a cluster of workstations using an ORB implemented in Java developed by IBM and available over the Internet. By means of web browsers, a user can access management information from any machine on the network. One difficulty they had was mapping the management functions

of SNMP and IIOP, because SNMP has limited resources when compared with the CORBA infrastructure.

The introduction of Java leads naturally to the Remote Method Invocation (RMI), which is an extension of Java that allows Java objects to be executed remotely. The RMI compiler generates a stub object for each remote interface, much like IDL. Unlike CORBA IDL which is language-neutral, RMI is focused on Java. There is no IDL needed between Java-based clients and servers, RMI can pass full Java objects as arguments and return values, not just pre-defined IDL data types. However, RMI can support multiple communications protocols including IIOP and this allows easy integration between Java components and CORBA components.

Sun defines JavaBeans as the component model for Java-based software, according to Monson-Haefel [3-16]. The model specifies how a component (or bean) can allow other components to discover its properties dynamically, and methods and events such that an application can be built by combining different components. A bean is a re-usable software building block, a pre-built piece of encapsulated application code that can be combined with other beans and with hand-written code to rapidly produce a custom application. Further, the JavaBeans model also specifies a set of APIs, so that a developer can develop an application that makes use of whatever services happen to be implemented on the target platform. It is this that makes Java / RMI applications easily portable from one platform to another. It is part of Java's 'write once, run anywhere' approach that has been demonstrated in uncountable numbers of Java applets being downloaded into web browsers on all types of platforms.

3.4.4 Web services

The trend is clearly towards use of the WWW to allow OSS servers and clients to communicate, in order to support distributed services that also use the Web. Here we have convergence – both the services and their OSS are tending to use distributed software components that communicate using the WWW, so it is worth giving a quick view of Web-based services, which are currently a very active research topic. Briefly, in order to be run remotely over the Web, a software component has to have three characteristics:

- it can describe itself, so that other components can understand its functionality,
- it can allow other components to locate it,
- it can be readily invoked.

Within the last three years, a number of technologies have matured and become world-wide standards to enable software components to be deployed across the Web. An impressive line-up of Companies is behind this standardisation, led by IBM and Microsoft. In March 2000, IBM publicly backed Microsoft's simple object access protocol (SOAP) that allows message-passing using XML. The major advantage of SOAP is that it allows applications to communicate using HTTP and standard web browsers and servers, therefore presenting no security problems with firewalls.

IBM and Microsoft then set about standardising an object description language that is now known as Web services description language (WSDL). At the same time, IBM and Ariba

started working on a solution to provide a means of discovering Web services and in September 2000, Universal Description, Discovery and Integration (UDDI) version 1 was announced. The development of SOAP, WSDL and UDDI have allowed the first phase of Web services to be introduced by Oracle, HP, Sun, IBM, BEA and Microsoft. For a good introduction to Web services and further information, see Muschamp [3-17].

3.5 Examples of COTS software

There is an abundance of integrated OSS software on the market that claims to have end-to-end carrier grade solutions. A brief survey of vendors has been made and figure 3-7 illustrates a generic integrated OSS stack with functions and some of these vendors identified at the various layers. A possible interface to a satellite earth station manager is shown. Note that, in the figure, business service, network and element management (i.e. the pyramid of figure 3-2) is contained in one box, since this is how much of the software is purchased. The central part of the software has a gateway for SNMP as shown, it is usually possible to configure gateways for CMIP or to write new gateways to handle proprietary element management protocols.

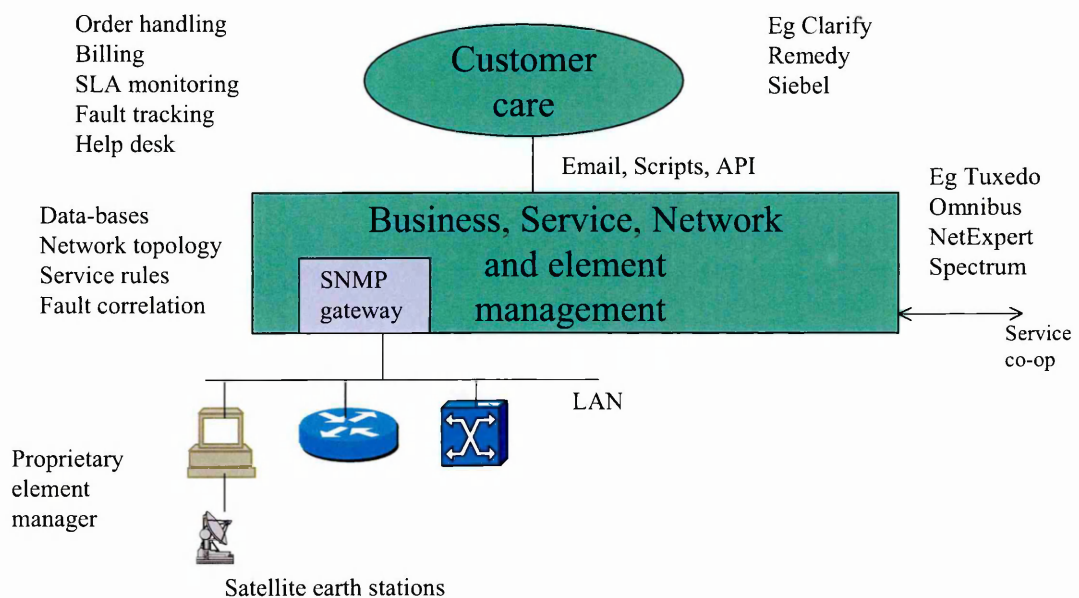


Figure 3-7 An integrated OSS structure

The Service co-op on the right-hand side of figure 3-7 is for invoking and managing services from third parties and interfaces with the service management layer in each OSS stack. This interface (called the X interface by the ITU-T) can use CMIP. It is used for business-to-business (B2B) communications as described by the Telemangement Forum (TMF).

Databases feature heavily in OSS architectures. Most use relational databases that are queried using a language such as Specification Query Language (SQL). Relational databases are fundamentally lists of data stored as tables. An alternative type of database, called deductive has been investigated in the context of network management [3-18], this allows a relation to be

defined in terms of another relation and use logic programming. For some types of query, this can lead to much simpler query structures.

3.6 Review of OSS industry models

There is a growing base of industry standards, common models, terminology and language relating to OSS in order that management software can have standardised functionality, protocols and interfaces so that it is usable in many different types of networks and platforms and hence be lower cost than bespoke systems.

In the 1980s the early attempts at standardisation pursued through the International Telecommunications Union (ITU-T (then the CCITT)) and the Internet Engineering Task Force (IETF) led to the development of means of interfacing to and instrumenting network elements, e.g. CMIS / P and SNMP. These point-to-point mechanisms are being augmented now by alternative means of interfacing, such as CORBA, OSS/J (J = Java) and Web Services that are able to interface more broadly between OSS applications.

The standards bodies that are most active in OSS developments are the ITU-T, IETF, Telecommunications Information Networking Architecture (TINA), The Parlay Group and the Telemanagement Forum (TMF).

The ITU is based in Geneva and writes recommendations, of which the Telecommunications Managed Network (TMN) is the most well-known in the field of OSS. The IETF meets in big venues around the world and writes RFCs that give recommendations on Internet related protocols. The TINA consortium (TINA-C) has constructed a framework for standardising interfaces and functionality of OSS software. The Parlay group writes APIs for Intelligent Network components. The TMF has constructed a business and technical OSS framework called the Telecoms Operations Map (TOM) and a second generation of this called the enterprise TOM (eTOM).

The activities of these standards bodies is now described in a little more detail, in order that a decision can be made on which standard business framework can be used in this study as a basis for evaluating the differences between satellite and terrestrial OSS systems.

3.6.1 ITU-T

The starting point for ITU OSS standards discussion is always the TMN, which was originally described by the M3xxx series of recommendations first published in the 1980s. Recommendation M3010 gives the principles for a TMN, that describes the interfaces and their functionality [3-4]. Recommendation M3013 defines horizontal layers in the structure [3-19], usually drawn as the famous TMN 'pyramid' shown earlier in figure 3-2.

Recommendation M3010 introduces the notion of a mediation device (similar to a proxy agent already discussed in section 3.4.2), which is a device that communicates with a TMN on one side using a TMN protocol and communicates with a network element on the other side using the protocol that is understood by the element. In this way, a TMN manager can manage non-TMN elements.

In terms of interfaces, M3010 defines the 'Q' interface, the 'F' interface and the 'X' interface. The Q interface has two sub-classes, the Q3 and Qx, where the Q3 is characterised by that portion of the information model shared between the operations system and the elements to which it directly relates. The protocol that runs over the Q3 interface conforms to the Open Systems Interconnection (OSI) model, the basic model of which is defined in recommendation X.200. Such an OSI protocol is CMIP as described earlier in section 3.4.2. The Qx is defined as a general interface between management application functions, but no implementations of this have been found, so it will not be pursued further.

The F interface is located between a workstation and the manager for human intervention and its functionality is defined in recommendation M3300. There have been no implementations of this found, so it will not be pursued further either.

The X interface is located between TMNs at the service layer and the protocols that run over it are given in recommendations Q811 and Q812, which are defined as being the lower layer and upper layers respectively of the protocol profiles for the Q3 interface. So, the conclusion is that the X interface is for service co-operation (sometimes it is called the X co-op interface) and it uses the Q3 protocol. Recommendations Q811 and 812 specify the use of Common Management Information Service (CMIS) and Common Management Information Protocol (CMIP) respectively. So, CMIP is specified both as an element manager protocol and as a protocol to support the X interface.

Recommendation M3400 describes the TMN management functions of Fault, Configuration, Accounting, Performance and Security (FCAPS) [3-20]. These are 'vertical' layers that cut through every layer of the pyramid in figure 3-2, so that every horizontal layer has processes to support each of the FCAPS.

Although FCAPS is described by the ITU-T, only two of these have been developed to the point where any functions have been standardised, these are Fault and Accounting. The development state of the others (C, P, S) is not sufficient to allow interworking between vendors. However, the basic TMN architecture has become a standard baseline for other standards bodies (notably the TMF) and for vendors of fully integrated OSS stacks.

Recommendation Q832-1 specifies the Q3 interface between a Service Node (SN) and a TMN, while Q832-2 specifies the Q3 interface between an Access Network (AN) and a TMN. The overall SN and AN interfaces to the TMN are called VB5.1 and VB5.2 interfaces respectively.

The Q3 interfaces specified are (1) between TMN Network Elements or Q-Adapters which interface to TMN Operations Systems (OSs) without mediation and (2) between OSs and mediation devices, as defined in Recommendation M.3010. The Q832 recommendation contains defined object classes for the Q3 interfaces, which will interface with the Network Management System using a protocol such as CMIP.

3.6.2 Parlay

The Parlay Group [3-21] is an open multi-vendor consortium formed to develop open technology-independent APIs enabling third-party service providers to run applications over networks. Basically it is about trust; the Parlay API allows applications from non-trusted environments to run over trusted networks as illustrated in figure 3-8.

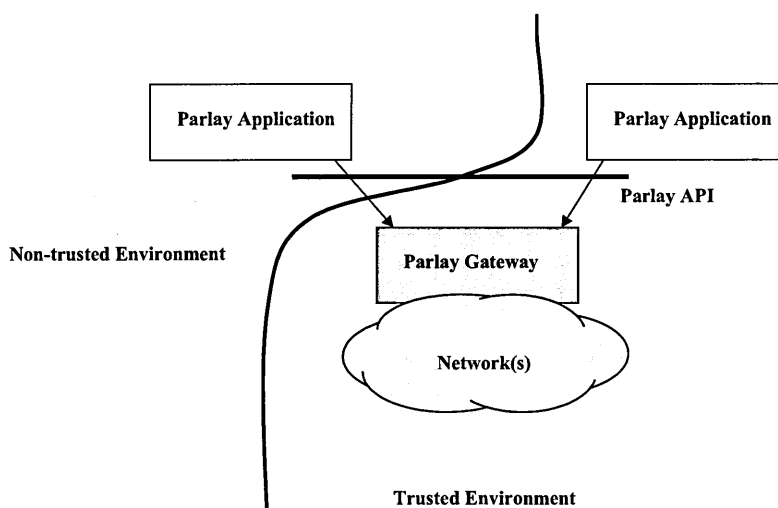


Figure 3-8 Role of Parlay API

A good overview of Parlay is given by Markwardt [3-22]. The Parlay Group is made up from several standards bodies including OMG, ETSI and 3GPP. There are several working groups, including ones for Java API realisation and Web Services. Parlay has published a white paper and several specifications for the interfaces, use cases and data definitions for policy management. Parlay is not considered further in this study since it operates at the application layer and is therefore independent of the type of network used.

3.6.3 TINA

The Telecommunications Information Networking Architecture consortium (TINA-C) was formed in 1993, for co-operatively defining a common management architecture. Some 40 telecommunications operators, telecommunications equipment and computer manufacturers joined the consortium. A Core Team, consisting of engineers from member companies, was created in New Jersey, USA, to join forces towards the achievement of a common goal.

TINA produced a document that explains the overall concepts and principles [3-23] that explains the reference points and sectors that they have defined; TINA is intended to be applied to all parts of telecommunications and information systems: for example, terminals (personal computers, etc.), transport servers (switching systems, routers, etc.), service servers (VoD, web, etc.) and management servers (authentication, billing, etc.).

TINA built upon TMN by clarifying the FCAPS functions and split Configuration into Resource Configuration and Connection Management. According to a report by Lewis from UCL, while TINA has produced relatively mature specifications, there is still a significant amount of research work to be undertaken before specifications and prototypes can be completed [3-24]. Lewis analyses service and network management problems using a combination of the TINA-C business roles and reference point segments and the TMF business processes. He shows how software solutions can be build from existing components, particularly in the area of event messaging. There is some good information about how to integrate software components to achieve service fulfillment, however the paper does not comment on the amount of management traffic overhead that such integration generates. TINA is not pursued further because its activity has reduced and it does not appear to have substantial support from service providers and software vendors.

3.6.4 The Telemanagement Forum (TMF)

TOM and eTOM

The TMF [3-25] is currently the most active and progressive of all the bodies developing OSS standards, consisting of telecommunications operators, OSS software vendors, integrators and equipment suppliers. It is therefore considered that the state-of-the-art in OSS reference models is contained in the TMF business process frameworks called the Telecoms Operations Map (TOM) and its successor the enterpriseTOM or eTOM.[3-1]. Both the TOM and the eTOM are briefly described. Figure 3-9 shows the TOM.

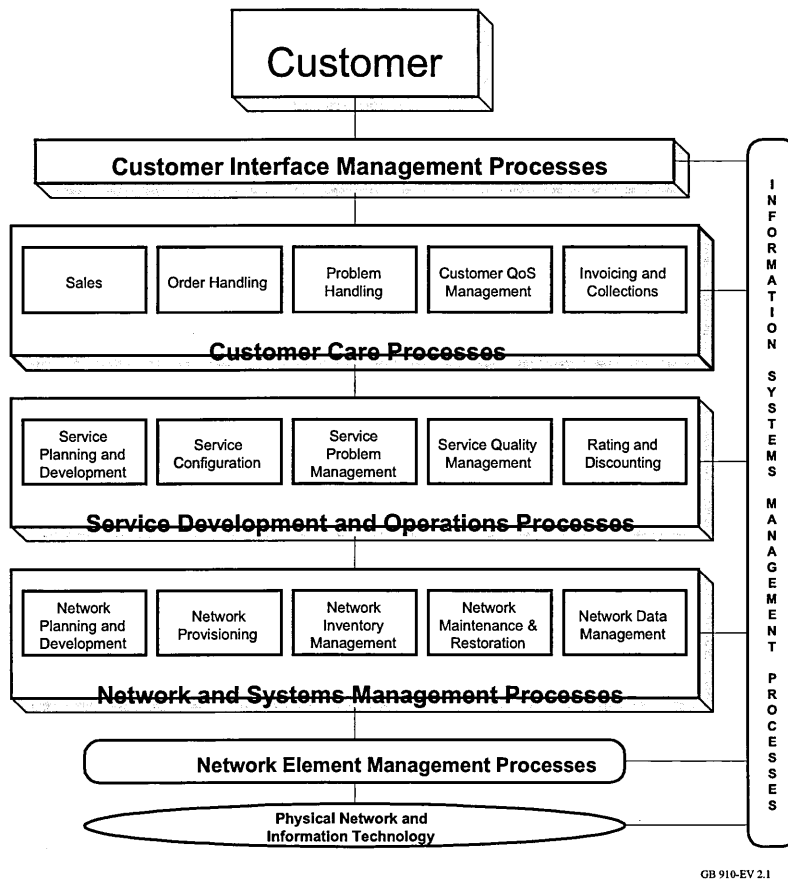


Figure 3-9 The TOM

Source: Telemanagement Forum

The TOM is also based on the ITU-T TMN architecture described in section 3.6.1. It has element and network management layers as in the pyramid in figure 3-2, but it splits the service management layer into a lower operations lifecycle layer and an upper customer lifecycle layer. The level of adoption of the TOM is such that some OSS software vendors include it in their literature to indicate which parts their software covers, for example NetBoss from Harris [3-26] and Agilent OSS [3-27].

Although it has been used for selling software and as a systems framework, the TOM is limited in that it was designed primarily for service providers (SP) and Network Operators (NWOs). The eTOM (enterprise TOM) overcomes this limitation by extending the applicability of the TOM to enterprises by developing its bottom layer: there is no network at the bottom, there is a more general 'resource management' and then the framework extends down to supplier / partner relationships.

Figure 3-10 shows the level 1 processes of the eTOM.

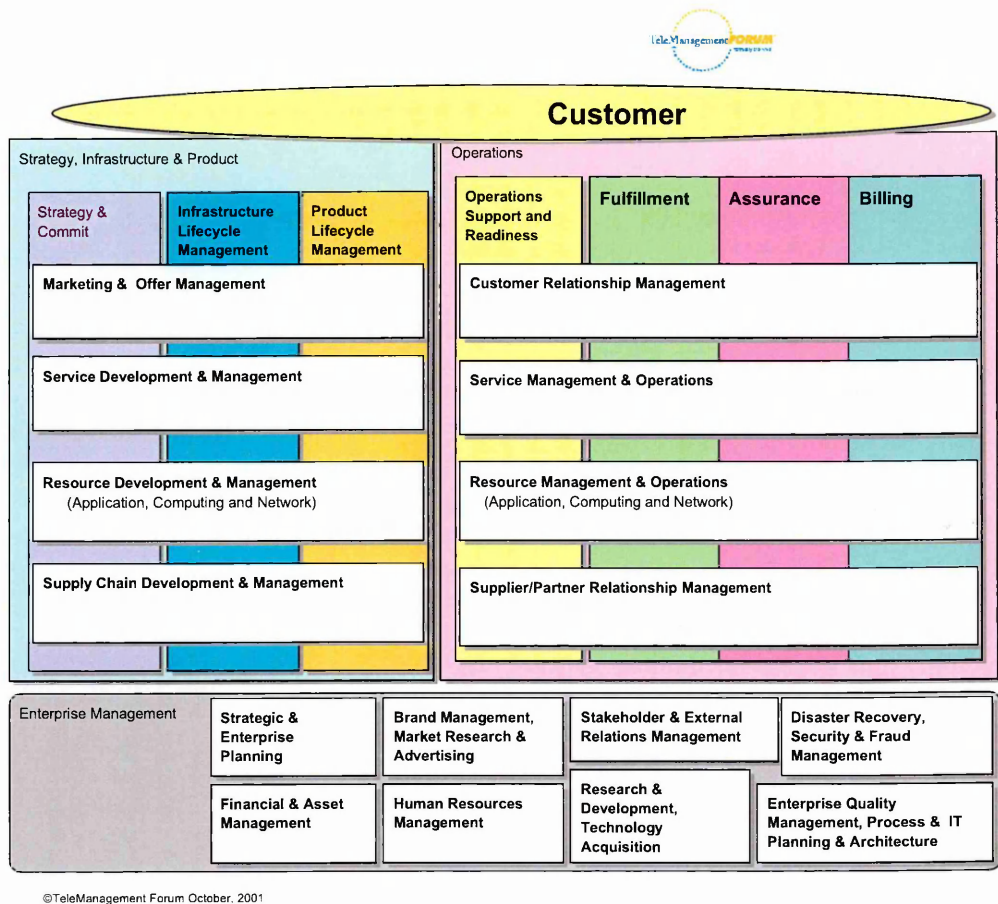


Figure 3-10 The eTOM level 1 processes framework

Source: Telemanagement Forum

The other development in moving from the TOM to the eTOM is the inclusion of vertical classifications of Strategy, Infrastructure and Product on the left, which contains processes for building a service, with Operations on the right.

Note that the eTOM does not have the ITU-T FCAPS as vertical layers, but these have been rationalised and re-classified as Fulfillment, Assurance and Billing (FAB). A fourth vertical layer is Operations Support and Readiness.

Of all the standards reference frameworks described in this section, the eTOM is the most developed and the most supported by software vendors, network operators and service providers. Therefore it will be used as the business framework context later in this chapter (in section 3.10), where its business processes will be analysed and those that are different to terrestrial networks will be identified.

New Generation OSS (NGOSS)

The distributed architectures described in section 3.4.3 are a big step forward from the bespoke fragmented OSS of a few years ago. However they are fundamentally ‘mesh’ systems

with pair-wise messaging and with this kind of topology the number of interfaces grows exponentially with network complexity. This also means that the difficulty in replacing OSS is increased, since each OSS was so tightly coupled to all the other systems it had interfaces with.

This led to the growth in the concepts of hub-based architectures, which are exemplified by the TMF's NGOSS approach [3-28]. In this approach, each OSS component interfaces with only one central component, the hub. The hub is therefore responsible for passing messages and handling transactions between the OSS components. The advantage of this approach is that, in principle, there only needs to be one interface per component, as opposed to the hundreds in the point-point approach. Of course, each of these interfaces has to support a wider range of processes, but nevertheless there are improvements in the cost of creating and supporting the interfaces.

The hub-based approach brings other advantages. With the point-point system, it is difficult to apply overall control to business processes, since each OSS component owns and controls a part of the overall process. Therefore changing or metricating the process becomes difficult and in some cases impossible. In a hub-based architecture it's the hub itself that is responsible for managing transactions, so it is also possible to use hub-based process (or workflow) engines to manage the overall process flow, with real benefits in terms of flexibility and control.

However, to make this all work as efficiently as it should, there are a number of other things needed to assist the integration. One of these is to work to a common information model. In a complex integration, the need for this is fairly self-evident – if one of the systems uses a different field size for customer name to another, or one uses 'zip code' as part of an address whereas another uses 'post code', then there will be problems passing information between them. Again, the TMF has done considerable work in this area with its Shared Information and Data (SID) model. This is starting to gain traction within the industry, with a number of providers of COTS components demonstrating SID compliance in their interfaces. As an example of recent commercial exploitation, Cutts points out that Vodafone are adopting the eTOM and SID framework for the OSS for their 3G network roll-out [3-29].

The approach taken by the TMF's NGOSS principles can be summarised as follows:

- Centralised process flow
- A common information model for shared data
- A common communications infrastructure
- Use of COTS components.

3.7 An overview of relevant OSS research

This section contains an account of current research activity in OSS to get an idea of the state-of-the-art into which the contribution of this study fits. There is some general OSS research papers at the start, chosen because of their emphasis on architectures. The papers towards the end of the section are on satellite-specific OSS.

Kajitani et al [3-30] explain the need for Network Management Systems and then they explain some problems in development of new systems. They assert that advancement in computer technology is too fast to be matched to the network environment, it is difficult to build a multi-vendor environment and it is difficult to cope with the complexity of network operations and the diversity of services. They further assert that NMS are fragmented across different kinds of networks and they suggest that, by using the Q adaptor function, a TMN can be extended a bit at a time. The adaptor is fitted to an existing NMS so that Q3 interfaced OSS can be connected. They have actually tried this and concluded that it is a reasonable approach in the early stages of integration. The authors mention that the Q adaptor software is difficult to maintain. Overall, they say the Q3 adaptor is not scalable – and that a better solution is to move managed objects into the target networks and use Q3 to manage them so it is all TMN compliant. They comment that the workload due to Q3 conversion can adversely affect overall performance.

Hewlett Packard suggest separating instrumentation from diagnosis and management [3-31] and actually this theme has been adopted by this study (in chapters 4 and 6) where measurement is considered only one part of the management process. HP argue that end-to-end QoS must use a distributed measurement system to align with applications and services that are also distributed. Services and applications have many components running on different networks and may have parts that run on Java applets embedded into web pages. The approach taken by the paper is that instrumentation exists in all of the parts in a lightweight scalable way, using code that is not dissimilar to the application code in each part, reporting to a distributed management system that performs measurements and diagnosis according to a set of rules.

The two papers above suggest that there are two different types of OSS traffic, one for measuring QoS parameters and the other for handling events, for example in the process of managing the QoS parameters. This suggests that the first type is in-band OSS traffic and the second type is out-of-band; this distinction is an important one and is discussed further in chapter 4 where QoS measurement methods and tools are reviewed.

Lewis et al [3-32] have considered integration across the service layer boundary between an ATM customer and service provider, using business processing to business role mapping. They used a combination of TMF process models and TINA reference points. Three components were created in CORBA, these were Subscription Management, Configuration Management and Network Planner which were used basically to enable a customer to place an order into the Service Provider for ATM connectivity. This is a rare example of a vertically integrated OSS stack to handle orders right through to provisioning. The paper stops there and does not consider the other vertical path that flows upwards: the flow of network fault information to the affected customers.

Turning now to some papers relating to satellite OSS, Villagra et al [3-45] developed a TMN X interface for a processing satellite network (EuroSkyWay), which included writing MIBs. The authors have developed a MIB using managed objects compliant with GDMO for a processing satellite network and used CMIP to manipulate values in it. The MIB is not given in the paper so the parameters they could manage are not known. They implemented a mediation device that communicated using an X interface (implemented using a reduced Q3

interface) to the manager containing the MIB. This means that the MIB can be manipulated from managers on other networks. On the element side of the proxy agent, a proprietary NWM protocol was used but they do not give any information about it. This paper points out a limitation of the TMN implementation as being the processing power of the manager workstations limiting the rate at which events can be handled. This accords with concerns in the paper referenced earlier by Kajitani that processing power in the manager limits the rate at which events can be handled. From this recurring theme it is deduced that management traffic and processes must place the minimum demand on processors; this theme is picked up and discussed in chapter 6.

Iera shows that traffic management can be used to mitigate the effects of latency on a satellite link [3-33]. Focussing on ATM real-time VBR traffic, he describes a technique that combines in-band signalling, distributed connection admission control and resource management. These, along with QoS, are negotiated at connection setup. This requires communication with the scheduler that controls the shared access and the technique is limited to ATM satellite networks that employ DAMA.

In the mobile satellite arena, the COST252 project is developing the satellite component of UMTS to provide broadband communications to mobile terminals according to Del Re [3-34]. COST252 are looking at traffic modelling and Network Management as part of this [3-35], where they analyse the satellite operating frequency bands and assess their suitability for this application.

Duressi et al assert that, over a satellite link carrying a mix of UDP and TCP services, the UDP will soak up spare capacity and that greater priority should be given to TCP streams [3-36]. This has also been found during this research; it is important to properly manage and regulate UDP traffic when both TCP and UDP are present on a network under congestion conditions. They simulated several TCP streams with 2 UDP streams over a satellite network with differentiated services. This is relevant because if SNMP is used, which is mandated to use UDP, the SNMP traffic must be managed in separate streams.

Zelnicker [3-42] confirms that a network management system should be integrated and cope with plug and play device detection and management (analogous to topology maintenance at the network layer) using a combination of in-band and out-of-band management traffic. In-band traffic accompanies the user data and is used for measurements, whereas out-of-band traffic has its own flows and is used for configuration and events. This distinction is picked up and discussed further in chapter 4.

The management of receivers in a satellite distribution network is described by Stein [3-43]. The paper focuses on the management of Integrated Receiver Decoders (for DVB transmissions) and how they can be configured while in use and recovered after outages caused by faults. There is no information on the overhead traffic required to achieve this.

A practical example of a integrated satellite system partially using TMN is from Tomasicchio who has written a paper on Skyplexnet [3-46], a distributed network management architecture to manage the Skyplex system on Eutlesat Hotbird 5 as mentioned in chapter 2. This architecture uses the FCAPS structure from the TMN. The paper focuses particularly on Fault,

Configuration and Performance management and constructs graphical representations of the network, making views available to users over the Internet.

Another example where FCAPS is specified for a processing satellite system is in a specification written by Intelsat for their Broadband Satellite System [3-47]. This system never progressed beyond the specification stage because Intelsat was forced to re-structure (see section 2.5.2). It is interesting that TMN is not mentioned anywhere in the specification document, but section 4 in the document has sub-headings entitled Fault, Configuration, Accounting, Performance and Security, which are the ITU-T derived classifications for the TMN.

Examples of practical satellite systems with fully integrated OSS stacks are rare. Two are known to be in the design stage but are not yet in service, they are Spaceway [3-37] and Skynet 5, a new military satellite system being procured from Paradigm by the UK MoD.

The element management of large earth stations has been implemented using COTS and customised software by some large service providers. These enable automated element management and remote control capability. One example is the Graphical Monitor and Control System (GMACS) from L3 [3-38] used by BT in the UK, and another is the Graphical Element Management System (GEMS) [3-39] used by AT & T to manage their satellite earth stations across North America. GMACS has one SNMP interface to the earth station elements and another for use with a higher manager and therefore can be used in cascaded fashion (this is a practical example of a cascaded element manager as shown in figure 3-3). GEMS collects alarms and status from monitored satellite communications equipment and presents them on a GUI. From the literature GEMS, unlike GMACS, does not appear to be capable of interfacing to higher-level managers.

VSAT systems are usually supplied with their own proprietary element management systems. An example is Nortel's SkyWAN [3-40] which interfaces with HP OpenView (a higher element manager) using SNMP, to provide configuration management, fault management and accounting. Another example is Linkway 2000 from Comsat [3-41]; this has a network management system that can supply statistics, topology and fault management to web-based clients.

Finally, a satellite network management system suited to rural networks is described by Satelink New Zealand [3-44]. This is an element management system that is tailored to demand-assigned schemes suitable for many rural users with little traffic each.

3.8 The value chain

3.8.1 Derivation of basic roles and actors

A stage of roles and actors is developed in this section in order to put OSS into a business perspective in terms of a value chain.

The stage is developed as a fresh approach from first principles. It is the start of another thread that will contribute to an analysis of the differences between satellite and terrestrial OSS later

in the chapter. The end results from this section have been validated by comparison with a white paper from the TMF on Value Chain Roles and Relationships [3-48].

Figure 3-11 shows a basic relationship between two entities.

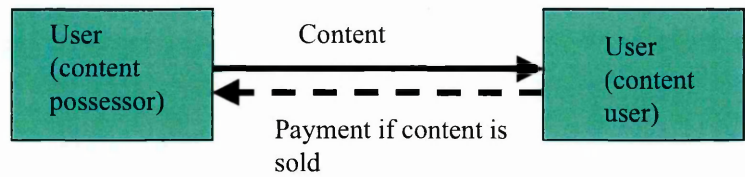


Figure 3-11 A fundamental diagram of information flow and payment

One entity wants to transmit information or content to another entity and it is immaterial whether it owns that content or not (it may be distributing the content on behalf of the owner). Some or all of the content flows from one user to the other and, in return, payment may or may not be made. Whether payment made or not is independent of whether content is ‘pushed’ or ‘pulled’. For example it is equally possible for the user to subscribe to a content possessor to have the content pushed and to pull content from the Internet without having to pay.

In the case of electronically stored information, one user wants the content and the other wants to deliver it. If it is to be paid for, the messages authorising and delivering the payment must be transferred. These functions can be performed by a new role, the Service Provider (SP). Figure 3-12 shows the content flow and any payment messages flowing through the SP (the actual money may or may not flow through the SP).

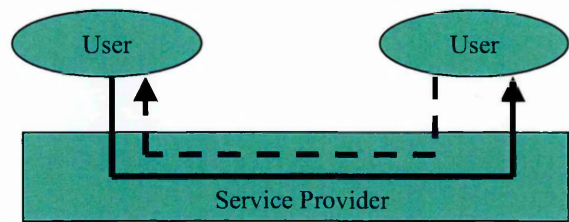


Figure 3-12 Making use of a Service Provider

The SP needs to be paid for providing service and we now move from the concept of content flow to that of service provision, because we want to move from a situation where the user is a customer of the content possessor to a different situation where both are now customers of the SP.

3.8.2 The service level agreement

Figure 3-13 introduces the notion of an Integrating Service Provider (iSP)¹, who is responsible for the storage, distribution and delivery of services, in accordance with contractual service

¹ The acronym iSP will be used for Integrating Service Provider, ISP will be used for Internet Service Provider.

level agreements (SLAs) with the both users or customers. SLAs exist between all the boxes in figure 3-13.

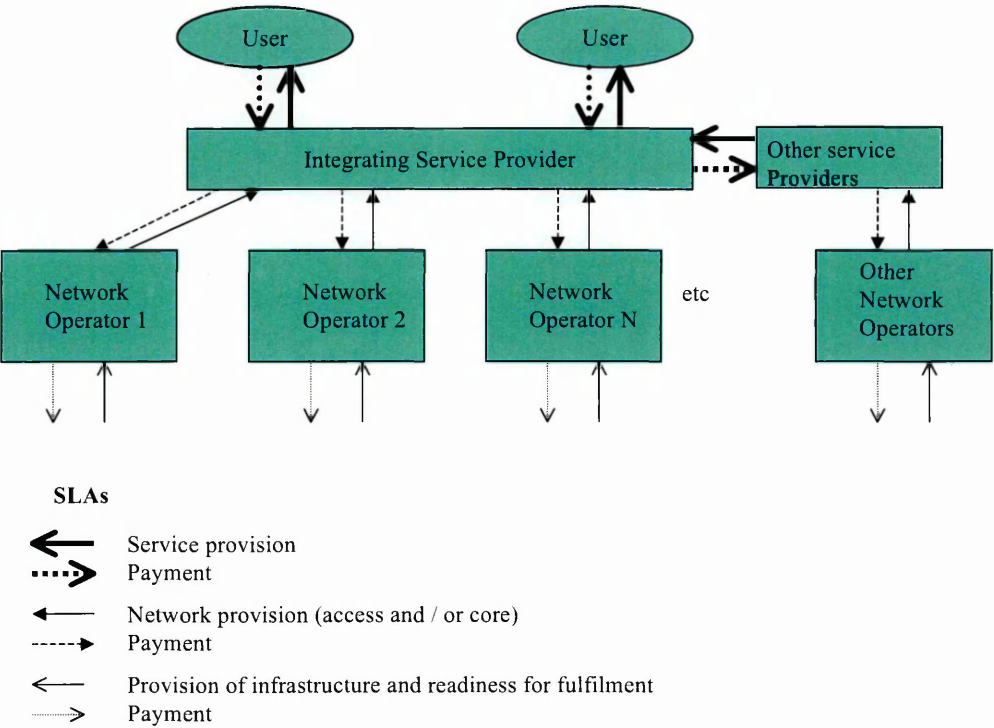


Figure 3-13 The use of an Integrating Service Provider to provide a service mix

Such SLAs contain specifications for the type of service (e.g. storage, changing the form of content, distribution and delivery), the Quality of Service (QoS) (e.g. availability, delay, delay variation, errors and bandwidth), security requirements, the amount to be paid and payment schedules (e.g. subscriptions and volume billing) and bilateral penalty conditions (e.g. for non-payment of bills or non-delivery of service as specified).

In order to fulfil the SLAs with its customers, the iSP will establish further SLAs with one or more network operators (of which one or more might be satellite) in order to transport the data. SLAs are also needed with other SPs, for example when a service is agreed with the customer must come from another SP (e.g. mobile or other specialist services like language translation). The reason for this is that iSPs try to become the one-stop shop for customers and offer bundled services that may in some cases force the iSP to buy services from other SPs in order to secure customer contracts. This is referred to by the TMF as ‘value networks’ [3-49]. The interface between the iSP and other SPs is business-to-business B2B, which is one area that is under development by the TMF in their work on the NGOSS (see section 3.6.4).

3.8.3 OSS and signalling flows

As stated in section 3.2, Operational Support Systems (OSS) is concerned with the Fulfilment, Assurance and Billing (FAB) of services and these are offered against SLAs.

Figure 3-14 shows that OSS functions (management plane) flow generally vertically in the business model (shown with solid lines) while signalling functions (control plane) flow generally horizontally.

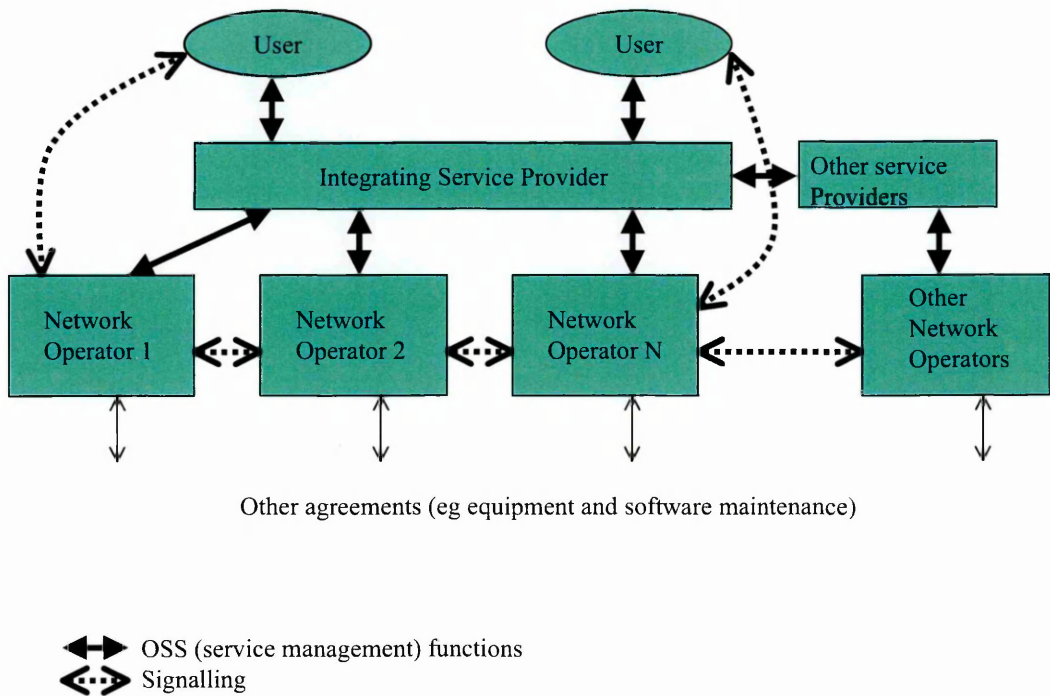


Figure 3-14 Showing OSS and Signalling paths

This demonstrates that the management plane and control plane are generally orthogonal to one another. Different NWOs are very reluctant to allow viewing, much less managing, any OSS parameters in each other's networks and in practice it is fraught with political and commercial difficulties. Network operators are very sensitive to other operators obtaining performance measures. Regarding QoS, requests for certain QoS parameter limits and negotiation can be made on the control plane for a particular service. Then measurement of the parameters and management of the agreed QoS is done on the management plane.

Examples of inter-network signalling that carry QoS parameter requests are the IMS in mobile networks, the ATM-forum NNI, routing metrics in BGP4 and supplementary services in CCITT No 7.

SLAs will be customised for different users, or different classes of user, because those QoS parameters that are important can vary. An example is end-to-end delay, which can be a fraction of a second for highly interactive sessions such as voice, or a few seconds for Internet access, or a few minutes for email. Depending on the type of application that the customer most uses, a short end-to-end delay time can be more or less important.

3.8.4 Management boundaries

The management responsibility of the NWO ends at the network termination point (NTP) at the user premises. This can be a phone socket on the wall. The management responsibility of

the iSP can extend beyond this point, say up to an ADSL modem or a satellite TV indoor unit. So, there is the requirement for the iSP to manage equipment and software at the user premises, via the NWO. Beyond these points, the equipment maintenance becomes the responsibility of the user.

The position of the management boundary and the ability of the user equipment to support OSS functions are important issues. Two example architectures will illustrate this, one with a one-way satellite link and one with a two-way satellite link.

Figure 3-15 shows a commercial system where an iSP (in this case Sky Television) is responsible for delivering content from the possessor (which can be a studio) to the user.

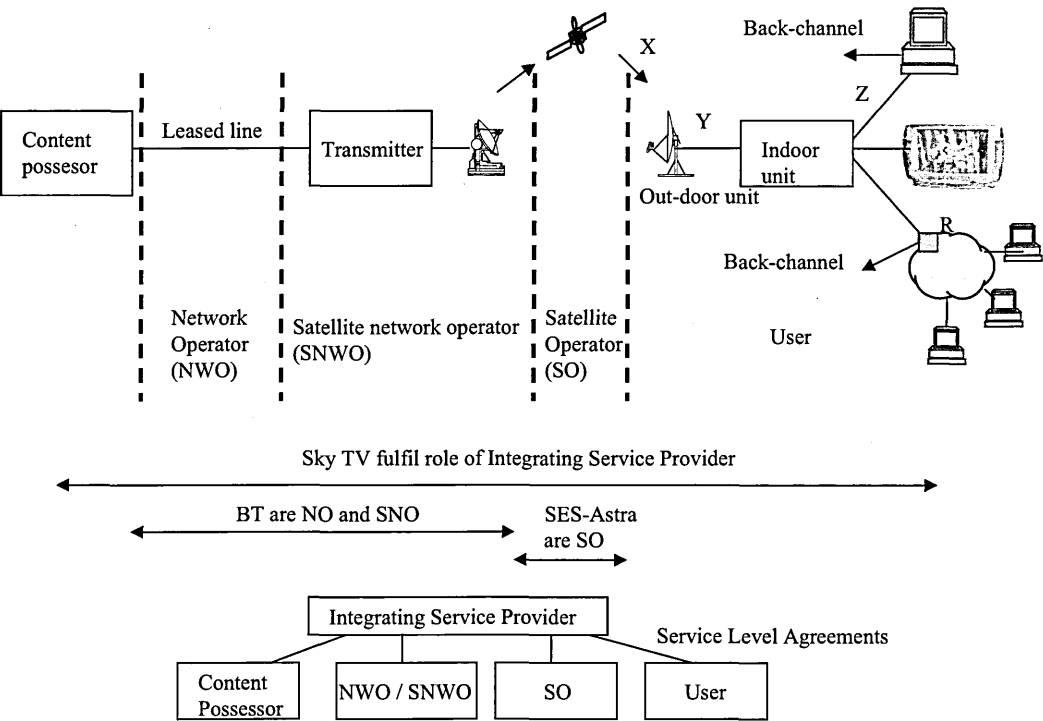


Figure 3-15 A practical example

Sky has several SLAs in place to achieve delivery of content to the user, through a value chain. The first layer in the value chain is represented by an SLA with BT to act as NWO and Satellite NWO (SNWO). In this case, the SNWO does not own satellites, so a second layer in the value chain is represented by an SLA between the SNWO and the Satellite Operator, in this case SES-Astra.

Figure 3-15 shows a 1-way satellite link carrying real-time video and audio streaming; this means that the receiver has one opportunity to receive it². All of the error protection must be done in the forward direction since there is no opportunity for link-layer ARQ or transport layer re-transmits to improve the reliability. This means that only the indoor unit has the ability to detect and report performance or fault problems to the iSP via a dial-up back-channel, perhaps with some hours delay. Figure 3-16 shows a two-way satellite system that allows the iSP to be informed much faster of such a problem, certainly within a few seconds. In this case the OSS traffic is tunnelled over the satellite link either from the indoor unit or from a router. The OSS traffic has to travel possibly in-band over a contended satellite channel, along with the customer's data.

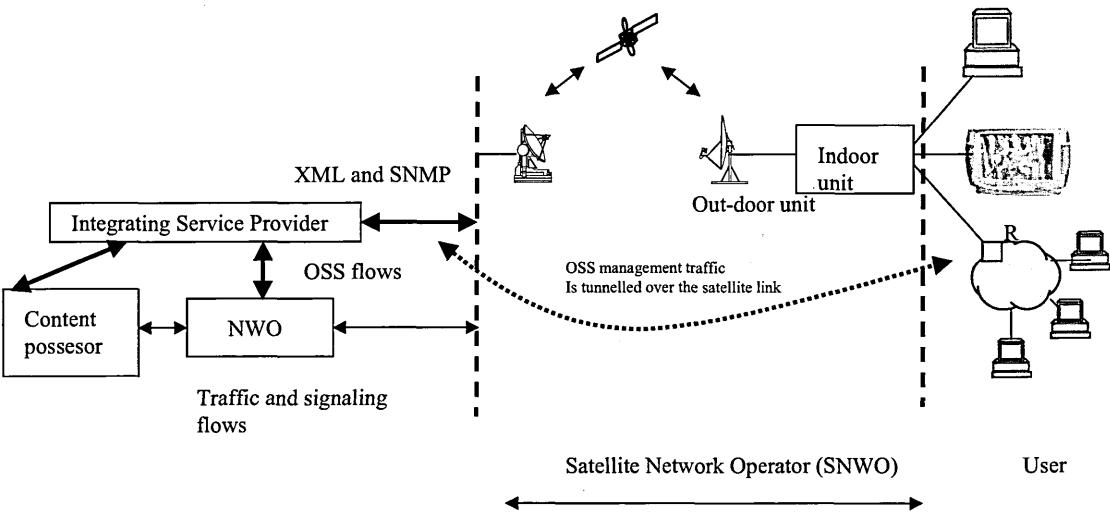


Figure 3-16 Tunnelling OSS traffic over a satellite link

The architecture in figure 3-16 was proposed by Spaceway [3-52] and by Intelsat for their broadband satellite design [3-53]. It is also used by a number of VSAT systems. Between the satellite operator network operations control centre (NOCC) and the iSP, XML is used for configuration and performance monitoring (so that these can be handled by a web browser) and SNMP is used for events. The speed at which it is possible to invoke performance management actions when a two-way link is present is addressed in chapter 6.

Both figure 3-15 and figure 3-16 show network options at the remote end of the satellite link, either a television, a PC or a small network connected using a router. In fact some indoor units have a router integrated with them (called integrated router decoders IRDs). The letters X, Y, Z and R on figure 3-15 indicate some possible management responsibility boundaries that the iSP may have.

There has been some industry papers and activity from standards bodies to identify these boundaries and define user equipment functionality to support the OSS processes necessary for fault, performance and configuration management.

² The Sky plus indoor unit can store programs and therefore is not strictly real-time. However there is still only one opportunity to receive the wanted programs.

The issue of management responsibility boundaries in the access portion of satellite links has been somewhat addressed from a service provider’s perspective in a paper by Wakeling [3-54]. The paper concludes that the SP wants to differentiate its services from the competition and hence needs control of the service delivery elements. The paper recommends the use of open standards for satellite access network design and promotes the FSAN (Full Services Access Network) initiative. It is worth exploring the work of FSAN a little, since they have performed some useful work in the definition of management boundaries.

The FSAN initiative was developed by CSELT, which is the research laboratory facility of Telecom Italia. FSAN introduces a term ‘service node interface’ (SNI), which is located at the interface to the core network, such as a gateway. The SNI is specified to be VB5 compliant and is known as a VB5 reference point.

The vision of FSAN comes from the copper and fibre terrestrial communications community, where, according to Quayle [3-55] the objectives were to reduce the number of different builds necessary for access line provision and to create a common requirements specification for an access system supporting a full range of narrowband and broadband services.

The Digital Audio Visual Council (DAVIC) and the ATM Forum also designed access reference models and specify the SNI to be VB5 compliant, but they call the interface different things from FSAN and from each other; DAVIC called it A4, ATM forum calls it ATM Network Interworking (ANI). The Universal Mobile Telephone System (UMTS) reference model calls this interface Iu but does not specify it to be VB5 compliant. Figure 3-17 shows the mapping of DAVIC, ATM-forum, UMTS and FSAN access reference models onto a satellite network, taken from Fitch and Fidler [3-56].

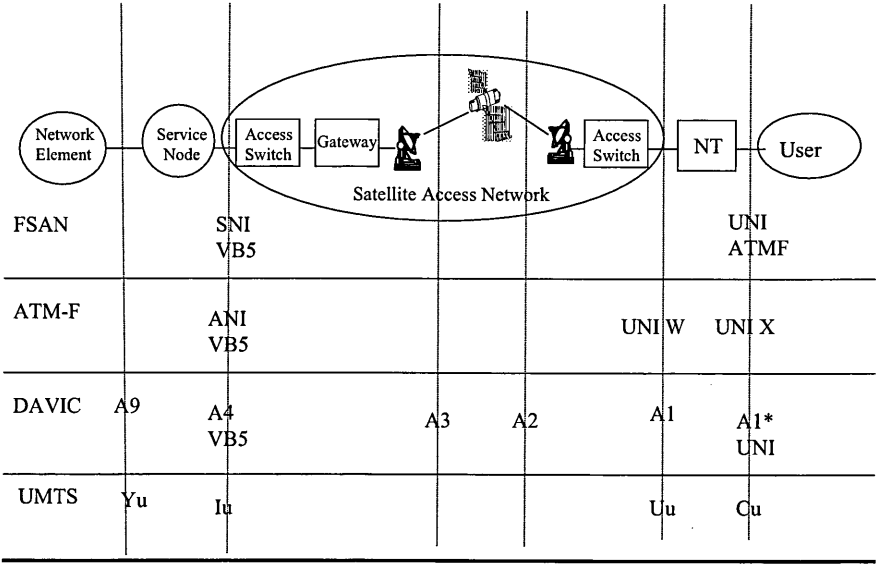


Figure 3-17 Mapping of FSAN, ATM Forum, DAVIC and UMTS Network Interface Nodes

The DAVIC forum was disbanded in 1999 and work on satellite (S)-UMTS is now centred on the ETSI SES group. Figure 3-17 shows that several forums have produced interface specifications for the management boundary at the network side of the service node, corresponding to point Z on figure 3-15.

Regarding the functionality in the user terminal to support OSS functionality, ETSI is currently writing standards for this and have issued a Technical Report written by the BSM working group within the SES committee [3-50]. This defines radio-dependent and non-radio-dependent processes and protocol layers in some detail, much of the input came from Hughes as a result of their experience with the Thuraya mobile satellite system.

The ETSI SES committee has been very active recently in publishing air interfaces for systems such as Thuraya, resulting from the EC issuing a directive [3-51] that public satellite service providers publish their air interfaces (point X on figure 3-15). This directive does not apply to the majority of VSAT systems because they are private.

3.9 Analysis of differences between satellite and terrestrial OSS processes

This section contains an analysis of OSS processes, within the eTOM framework developed by the TMF. In section 3.6.4 the eTOM was chosen as the most relevant business process framework on which to conduct this analysis.

3.9.1 The eTOM level 1 processes

The eTOM is a collection of business processes at different levels; 1 and 2 which are medium and high levels of detail respectively. In order to focus down onto topics for further study, the method used was to identify those level 2 processes that are different between satellite and terrestrial links. From these differences, a further study into how best to progress satellite OSS will be made. The point of departure is the eTOM level 0 processes given earlier in figure 3-10, repeated as figure 3-18.

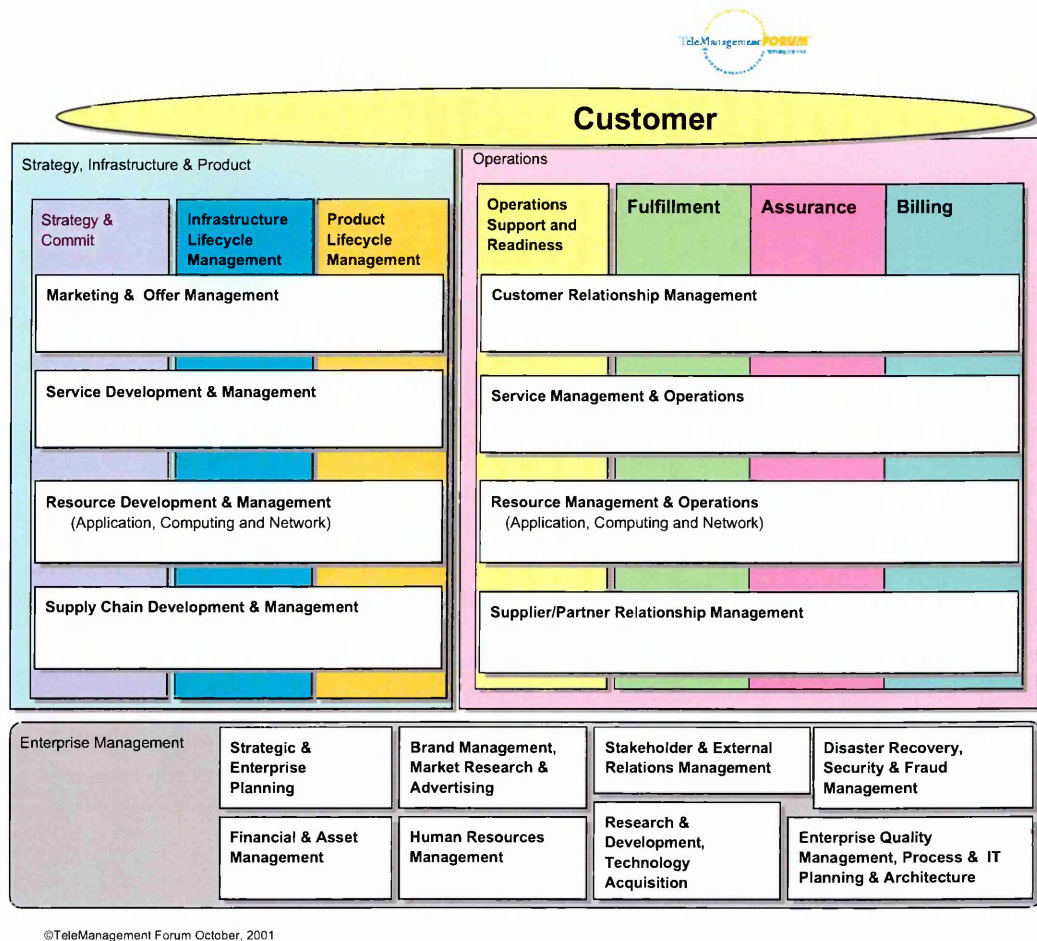


Figure 3-18 eTOM level 1 processes

Source: Telemanagement Forum

The eTOM Business Process Component framework represents the whole of a service provider's enterprise environment. The eTOM can be viewed as having three major process areas that contain sets of level 1 process groups. These process areas are

1. Operation Area Process Components, which realises the core operational management processes for the day to day activities experienced by customers and are organised in 3 primary vertical groups (the fourth is concerned with support and management of the other three). They are the FAB groups plus Operations Support and Readiness as already introduced but expanded here:
 - Fulfillment i.e. taking orders, and providing service via whatever service and network configuration activities are required,
 - Assurance i.e. dealing with customer problems, performance and service level agreements,
 - Billing i.e. billing and collection management, service specific rating and resource usage data collection and analysis,

- Operations Support and Readiness, i.e. customer relationship management, service management and operations and resource management and operations.

There are also 4 horizontal layers that broadly map to the TMN layered model (see figure 3-2) which span the Operations and Strategy areas:

- Customer Management deals with the range of interactions with customer and the lifecycle of service instances as delivered to customers,
 - Service Management deals with the lifecycle of services offered by the service provider,
 - Resource Management deals with the lifecycle, from planning and building through to operation, of resources (Applications, Computing and Network) operated by the service provider and which form the basis of the services as delivered,
 - Supplier Management deals with the lifecycle of supplier relationships whether suppliers are equipment providers or other service providers.
2. Strategy Infrastructure and Product area Process Components. These deal with back office processes that plan business direction, manage the lifecycle of old and new products and develop, plan and deploy new infrastructure essential to the efficient operation of the Operational Processes.
 3. Enterprise Management area, covers Process Components necessary in all industries (sometimes called horizontal industry processes). Examples include HR planning, Project Planning and Management processes, processes related to general industry needs, e.g. treasury functions, legal and accounting functions.

The TMF have models which address other aspects of OSS i.e. systems integration, data and interfacing and the eTOM provides the high level context for all of these. In this study we are only concerned about the right-hand side of figure 3-18.

3.9.2 Level 2 processes

There is no overall diagram for level 2 processes; these are described using much text and many small diagrams in the eTOM Business Process Framework document [3-1]. The relevant level 2 processes are described next and then are summarised in a table that points out the differences between satellite and terrestrial links. The entire list of level 2 processes in the eTOM, of which there are several 10s, has been short-listed by initial filtering to produce candidate processes that may be relevant to satellite. These are listed below with the name of the process given and a brief summary of each.

- 'Marketing Fulfilment Response' processes are responsible for distribution of marketing collateral (i.e., coupon, premium, sample, toys, fliers, etc.) directly to a customer, tracking the lead, etc. These processes include campaign management activities from lead generation to product and literature fulfilment and lead hand-off for selling.

- 'Selling' processes include working to manage prospects, to qualify and educate the customer and to create a match between customer's expectations and enterprise's products and services and ability to deliver. These processes also manage response to customer Request for Products (RFP)s.
- 'Service Configuration and Activation' processes encompass the installation and configuration of the service for customers, including the installation of customer premises equipment. They also support the re-configuration of the service (either due to customer demand or problem resolution) after the initial service installation. This can include modifying capacity and reconfiguring in response to requests from other providers.
- 'Resource Provisioning and Allocation to Service Instance' processes encompass the configuration of resources, and logical resource provisioning for individual customer instances. This involves updating of the Resource Inventory Database to reflect the resource being used for a specific customer.
- 'Supplier/ Partner Interface Management' processes manage the contacts between the enterprise and its current or future suppliers/partners for products or services. These processes are basically contact management and tracking processes. These S/P Interface Management processes interface with the CRM process of Customer Interface Management.
- 'Customer QoS/SLA Management' processes encompass monitoring, managing and reporting of Quality of Service (QoS) as defined in Service Descriptions and performance in relation to Service Level Agreements (SLA) for service instances, and other service-related documents. They include network performance, but also performance across all of a service's parameters, e.g., % completion on time for order requests.
- 'Service Quality Analysis, Action and Reporting' processes encompass monitoring, analysing and controlling the performance of a service perceived by all customers of a service class. These processes are responsible for restoring the service performance to a level specified in a SLA or other service descriptions as soon as possible.
- 'Resource Restoration' processes deal with resolving problems with groups of resources, so that the resource capability is restored.
- 'S/P Performance Management' processes track, measure and report supplier and partner performance. These processes interface with the supplier's CRM process of Customer QoS and SLA Management.
- 'Resource Problem Management' processes are responsible for the day-to-day management of problems with groups of resources (resource classes), and ensuring that the resources are working effectively and efficiently. The objective of these processes is to proactively deal with resource problems, before complaints are received about affected services.

- 'Resource Data Collection, Analysis and Control' processes encompass the collection of usage, network and information technology events, including resource information, for customer usage reporting and billing. This also includes analysis of the collected information to understand the impact on resource performance, and based on analysis of this, installs controls to optimise this performance. These processes collect and format data for use by many other processes within the enterprise. Based on the Product strategy for the enterprise, the Product & Offer Portfolio Capability Delivery processes manage the delivery and build of new or changed product capabilities e.g. 3G mobile telephony national network roll-out.
- 'Service Development and Retirement' processes are project oriented in that they develop and deliver new or enhanced services. These processes include process and procedure implementation, systems changes and customer documentation. They also undertake rollout and testing of the service, capacity management and costing of the service. It ensures the ability of the enterprise to deliver the service to requirements.
- 'Service Performance Assessment' processes assess whether the Service Development and Management processes are meeting their goals of delivering changes and improvements to the Service Management & Operations Processes.
- 'Financial Management' processes manage the financial aspects of the enterprise such as Treasury, Banking, Payroll, Financial Planning, Auditing and Accounting Operations functions, e.g., Accounts Receivable and Payable. These processes are accountable for the financial health of the enterprise, managing cash flow, auditing for compliance to financial and expense policies, etc.
- 'Real Estate Management' processes manage all aspects of corporate real estate such as planning for future needs, purchase and lease of real estate, build-out, maintenance and disposal.
- 'Procurement Management' processes procure the necessary goods and services for the non-production needs of the enterprise. For the most part, Procurement Management does not deal with the purchasing of goods and services required for the infrastructure for delivering products and services to customers.
- 'Regulatory Management' processes ensure that the enterprise complies with all existing government regulations. Additionally, this process is responsible for legally influencing pending regulations and statutes for the benefit of the enterprise and to inform the enterprise of potential consequences of pending legislation or regulations. In addition, these processes are responsible for tariff filings as required.
- 'Disaster Recovery & Contingency Planning' processes are responsible for setting corporate policies, guidelines, best practices and auditing for compliance by all entities for disaster recovery and restoration, including contingency planning. These processes are accountable for ensuring disaster recovery plans are in place and tested to ensure continuity of service to customers and revenue to the enterprise in case of a disaster.

- ‘Security Management’ processes are responsible for setting Security Management corporate policies, guidelines, best practices and auditing for compliance by all entities. Security Management addresses internal and external sources of security violations. These processes strongly interact with Fraud Management and have common elements and information services and communications specific elements. Security processes are implemented at many levels of the enterprise and at the user, system, network, etc. levels.
- ‘Fraud Management’ processes are responsible for setting Fraud Management corporate policies, guidelines, best practices and auditing for compliance by all entities. Fraud Management addresses internal and external sources of security violations. These processes strongly interact with Security Management and have common elements and information services and communications specific elements. Fraud processes are implemented at many levels of the enterprise and at the user, system, network, etc. levels.
- ‘Resource Management and Operations Readiness’ processes ensure that application, computing and network resources are ready in support of provisioning and maintenance of resources to provide service, and of the RM&O processes. This includes the configuration of the resources, and logical resource provisioning to be able to support specific service types. These processes are also responsible for supporting new product and feature introductions and enhancements in development and/or review of processes and methods and procedures, as well conducting Operations Readiness Testing (ORT) and acceptance. Readiness processes develop the methods and procedures for the specific process and function and keep them up-to-date, including making improvements. Before Operations accepts a new product, feature or enhancement, operations readiness testing is required that is ‘hands off’ from the developers. After fixes identified in operations readiness testing are completed, these processes accept the new or enhanced product and features in full-scale introduction or general availability.

3.9.3 Identification and prioritisation of processes relevant to satellite

The level 2 processes above and their hierarchical relationship to level 1 and level 0 processes are listed in the first three columns in table 3.1. The third column gives the differences between satellite and terrestrial networks as relevant to each process. The final column shows the ranking that has been placed on the differences according to how significant the difference is and this ranking will be used for further sifting.

The ranking was decided by considering how much the business process impacts on the management overhead traffic and hence on the satellite channel capacity. For instance, ‘Marketing response’ is different between satellite and terrestrial, but for business reasons and so does not affect the management traffic, therefore this is ranked as ‘3’. On the other hand ‘Service management and operations’ implies a high level of management traffic and so is ranked as ‘1’.

	Level 1 process	Level 2 process	In what way different	Ranking (1-3, 1 = most relevant)
Fulfilment	Customer Relationship management	Marketing response	Fast connection time and flexibility.	3
		Selling	Specialist technical knowledge	3
	Service management and ops	Service configuration and activation	In-band configuration and activation. Includes provision of CPE.	1
	Resource management and ops	Resource provisioning and allocation to service instance	Signalling and contention management. Database updating of resources allocated to each customer	1
	Supplier / Partner	Interface management	Bundling of services	3
Assurance	Customer Relationship Management	QoS / SLA	Different QoS parameters are important to different customer types. SLA monitoring is different because of different service set and penalties / pre-emptive service offerings	1
		Service quality analysis, action and reporting	Measurement method depends on satellite network configuration	1
	Resource Management and Operations	Resource restoration (network)	Restoration depends on satellite sparing policy of SNWO and SO. Topology, topology discovery and rules are different for satellite	1
		Data collection (network)	Measurement method depends on satellite network configuration	1
	Supplier / Partner	Performance management	QoS parameter allocations different due to latency and broadcast nature of satellite	3
Billing	Resource Management and Operations	Resource data collection (network)	Adaptive pricing and routing	1
Operational Support and Readiness	Customer Relationship Management	Operations readiness	Network resources available. Appropriate GoS when not fully contended	1
	Service Management and Operations	Service Support and processes	Specialist processes and help-desk knowledge	3
		Service Readiness	Trials different for satellite ?	3
	Resource Management and Operations	Resource Support and processes	Reference models different for satellite	3

		Resource readiness	Constellation readiness. Earth stations. CPE (or handset) distribution	2
Infrastructure lifecycle	Marketing and offer management	Product and offer capability	Lifecycle of satellites, ground equipment. Product lifecycle, disruptive products and infrastructure	2
Enterprise Management	Financial and Asset management	Financial Management	Investment and cost of service launch. Payback periods, EBIT	2
		Procurement	Of satellites, ground equipment, backhauls and network interconnect. OSS software. Note that the eTOM says procurement is not for goods in order to offer services.	3
		Real estate	Earth station sites, CPE mounting	3
	Stakeholder and External relations	Public and community relations	Antenna siting, satellite disposal	3
		Regulatory	Type approval of equipment. Licenses to transmit and to offer service. Satellite slot filing, spectrum management and co-ordination.	3
	Security and Fraud	Disaster recovery	Recovery and restoration	3
		Security management	In broadcast environment	1
		Fraud management	In broadcast environment	3

Table 3-1 eTOM Layer 0, 1 and 2 processes that are different for satellite and terrestrial systems

3.9.4 Mapping of processes onto satellite architectures

Having defined the eTOM level 2 processes that are most relevant (the ones marked with a ‘1’ in table 3-1) it is necessary to put them into context. Figure 3-19 contains a set of satellite link type options, this figure is repeated from section 2.5.1 for convenience.

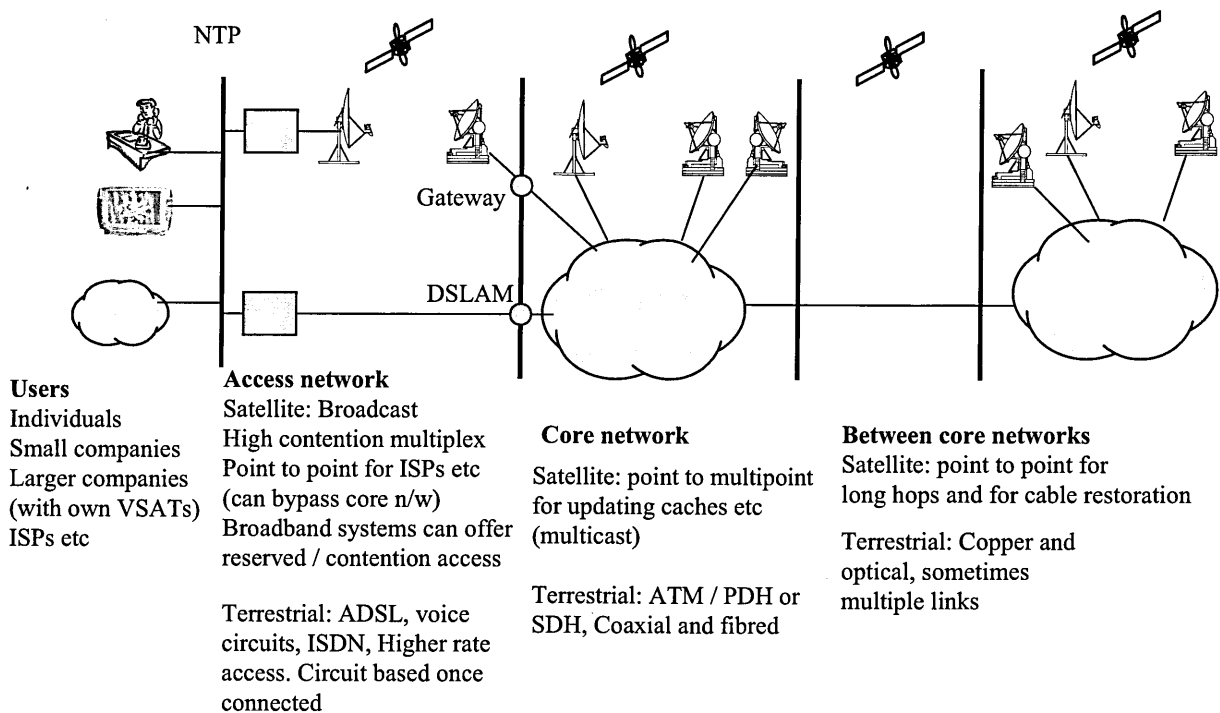


Figure 3-19 where satellites are used

Figure 3-19 shows that the places where satellites are used are in the access network, intra core network and between core networks. The between core networks role will not be considered further because it has already been established that these are low-volume high-value links, which generally operate without contention and hence would have limited benefits from the findings in this study.

Table 3-2 shows how the high-priority eTOM layer 2 processes map onto the remaining link types, the Access and Core network. The Access has been divided into one-way and two-way satellite links, where one-way would typically serve the consumer market and two-way would serve the corporate market.

eTOM layer 2 process	Access (one-way) Possibly contended, in-band management. Individual or groups of users, companies including ISPs. One-way, asymmetric or symmetric. Star configuration	Access (two-way) Corporate LAN interconnect, private and possibly secure communications networks Two-way, asymmetric or symmetric. Star or mesh configuration.	Core network Delivery of web pages and broadcast items to caches at edges of network (such as tier 3 exchanges or on user PCs). One-way, not contended. Star configuration
Service configuration and activation	<i>Event traffic, possibly contended, to configure user terminals. Hybrid / two-way is different. CPE logistics including training fitters, site clearance, licensing and planning permission.</i>	<i>Event traffic, possibly contended. Interface to corporate OSS</i>	Out of band, not contended
Resource provisioning and allocation to service instance	Management of congestion and restoration of service. Database updating of resources allocated to each customer.	<i>As with access, but on wholesale scale to corporate</i>	Schedule based
QoS/ SLA (Including allocation of QoS parameters across network operators) Refers to instances of services	Uses OSS link between iSP and SNO for allocations. QoS measurements using real-time overhead. <i>Possible reporting traffic from user terminals</i>	Uses OSS link between iSP and SNO for allocations. QoS measurements using real-time overhead. <i>Reporting is to iSP OSS and then a view to corporate OSS.</i>	Delay not an issue, therefore real-time not used. Confirmation of file transfers is needed
Service quality analysis, action and reporting. Refers to service class. Post processing and reporting	Out of band management traffic	Out of band management traffic	No impact
Resource restoration (network) and data collection	<i>Diversity for larger users, using alternative earth stations, transponders or satellites. Managed by prediction (eg eclipses) or ad-hoc – high priority events (due to faults, bad weather, cost and availability)</i>	<i>As access</i>	<i>As access</i>
Operations readiness	Network infrastructure, including computing resource, databases, fault clearance processes etc. Training of help-desk.	<i>As access</i>	<i>As access, but without QoS management</i>
Security	<i>Key and certificate distribution. Authenticated web sites, trusted firewalls and good behaviour when links disappear</i>	<i>As access</i>	<i>As access</i>

Table 3-2 Mapping of rank = 1 eTOM Level 2 processes mapped onto satellite link types

3.9.5 Identification of areas for further study

The *italic* font in table 3-2 indicates those processes that (a) are most different between satellite and terrestrial and (b) affect the traffic flow at layers 1 and 2 of the OSI stack (Physical and Link) which is where satellites generally operate. The ***bold italic*** font indicates those processes that are identified as directly affecting the management overhead traffic, so that further development of these would directly improve the cost-effectiveness and appeal of communicating by satellite.

The areas identified for further study are therefore:

1. Event traffic to configure user terminals
2. QoS measurements using real-time overhead
3. Management of congestion and restoration of service, including high priority events due to faults, bad weather, cost and availability.
4. Service quality monitoring and analysis

Expanding these a little:

1. Configuration and activation of user terminals involves OSS traffic in-band to remote terminals, in order to set up service parameters and authenticate services to users. This is overhead traffic that is not very delay sensitive, for example several minutes delay seems reasonable for adding a new service such as email. Service parameters might be:

- Routing tables and firewall distribution and configuration
- Performance enhancing proxy distribution and configuration (e.g. for TCP)
- User authentication information (e.g. for groups of users)
- Setting service profiles (eg turning services on and off, allocating traffic management parameters such as committed access rates, bandwidth, delay and delay variation apportionment)

2. QoS measurements using real-time overhead. This is for SLA reporting and for informing the management systems quickly when there is a problem. An efficient means of measuring QoS parameters is essential for satellite links where capacity is expensive and power and bandwidth are limited. It must also be economical with processor power, since it is unlikely that much will be available at the remote ends of satellite links.

3. Management of congestion and restoration of service. This is urgent management event traffic that has to act quickly to restore QoS to minimise the impact on SLAs. Such actions can include choosing the best and alternative routes across available networks, increasing bandwidths and allocating larger buffers.

4. Service quality monitoring and analysis. This is to perform SLA policing and for trend analysis for policy and strategy decisions, for example planning network upgrades.

3.9.6 Conclusion of evaluation

In conclusion, the key differences between satellite OSS and terrestrial OSS that impact the amount of management traffic are in the areas of service configuration, QoS / SLA

management and resource restoration. Of these, the service configuration is the least difficult since this is not time critical and can take place during quiet periods. Next in order of difficulty is the resource restoration, since this is time-critical and high priority; it is recommended that this is managed by programming queues to prioritise this traffic.

The most difficult is the real-time measurements of QoS parameters, which must be done real-time and use the minimum overhead traffic due to the high cost and limited power and bandwidth on a satellite link. It is therefore this issue that is taken forward for deeper study with the question 'which QoS parameters to manage and how to best measure them in real-time'. Chapter 4 addresses this question.

There are other differences in business processes such as marketing, finance and asset management, which are listed in table 3-1 that do not impact the overhead traffic and so are not discussed further.

3.10 References for chapter 3

- [3-1] Telemanagement Forum document number GB921. eTOM, *The Business Process Framework for the Information and Communications Services Industry*. Member evaluation version 2.5, October 2001.
- [3-2] Oliver Daures. *OSS Solutions for IP: The new challenges*. Compaq White Paper, Version 1.0, December 2000.
- [3-3] Tapio Kaijanen. *NGN & 3G, ETSI bids for the future*. ETSI presentation to INET2002.
- [3-4] *Principles for a Telecommunications management network*. ITU-T Recommendation M3010, May 1996. Pages 32 - 34
- [3-5] *PanaVue Management Platform for the Promina 800 series network*. Taken from the Promina 800 product guide, chapter 10.
- [3-6] *Information technology – Open Systems Interconnection – Common management information service definition*. BS ISO/IEC 9595 : 1991, plus Cor. 3.
- [3-7] *Information technology – Open Systems Interconnection – Common management information protocol*. BS ISO/IEC 9596-1 : 1991. Part 1 : Specification.
- [3-8] *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*. BS ISO/IEC 8824 : 1990.
- [3-9] *Information technology – Open Systems Interconnection – Structure of management information – Part 4 : Guidelines for the definition of managed objects*. BS ISO/IEC 10165-4 : 1992.
- [3-10] Walt Dymek. *SNMP versus CMIP Network Management*. Datapro Communications Equipment and Services, March 1994.
- [3-11] David Breitgand, Danny Raz, Yuval Shavitt. *SNMP GetPRev: An efficient way to browse large MIB tables*. IEEE Journal on Selected Areas in Communications, Vol 20, No. 4, May 2002. Pages 656- 667.
- [3-12] Andrew Fidler, Mike Fitch, Guillem Hernandez, Ken Fowler. *EC project VANTAGE, Trial 3 report*. BT Report number SSTA 0168. 29th September 1998.
- [3-13] YK Tang. *Distributed Network Management Software Study*. Lockheed Martin, 2001.

- [3-14] Djalalian et al. *Unified Web-based Network Management Based on Distributed Object Orientated Software Agents*. Proceedings of SPIE, Volume 4909 Network Design and Management. September 2002, pages 51-59.
- [3-15] Barotto, de Sousa Westphall. *Distributed Network Management Using SNMP, Java, WWW and CORBA*. Journal of Network and Systems management, Vol 8 No. 4 2000. Pages 483 – 497.
- [3-16] Richard Monson-Haefel. *Enterprise JavaBeans*. ISBN 0-596-00226-2. O-Reilly and Associates, 2001.
- [3-17] Paul Muschamp, *An Introduction to Web Services*, BT Technology Journal, Vol. 22 No. 1, January 2004, pages 9 – 18. (The whole of this BTTJ is about Web Services).
- [3-18] N Sharda, R Fatri, M Abid. *Applying Deductive Database Technology to Network Management*. ACM Sigcomm Computer Communications Review, pages 42 – 54.
- [3-19] *Considerations for a Telecommunications Management Network*. ITU-T Recommendation M3013, February 2000.
- [3-20] *TMN Management Functions*. ITU-T Recommendation M.3400 (10/92)
- [3-21] The Parlay group. See <http://www.parlay.org/specs/index.asp> Link Checked 1st December 2004
- [3-22] Gert Markwardt. *Parlay Standardisation*. Sun / Seimens Parlay Developer Workshop, April 11th 2002.
- [3-23] Martin Chapman and Stefano Montesi. *Overall Concepts and Principles of TINA*, Document TB_MDC.018_1.0_94. 17th February 1995. Also see <http://www.eurescom.de/~public-seminars/1997/DOT/slides/Session2/09Gavras/> Link checked 1st December 2004.
- [3-24] David Lewis, University College London. *TeleManagement Forum Business Process to TINA-C Business Role Mapping*. See <http://www.cs.ucl.ac.uk/research/flowthru/content/bmp-role-map/bmp-role-map.html> Link checked 1st December 2004.
- [3-25] The Telemanagement Forum. See www.tmforum.com. Link checked 1st December 2004.
- [3-26] Harris Network Management and OSS: *NetBoss*. See http://www.networksupport.harris.com/Contact/Netboss_sales.html Link checked 1st December 2004.
- [3-27] Rudiger Sellin. *Agilent als Anbieter von OSS-Komplettlosungen (in German)*. Comtec Magazine June 2002.

- [3-28] Telemanagement Forum document number TMF051. *New Generation Operational Support System, Part 1, Business Case*. September 2000.
- [3-29] Roger Cutts. *Use of eTOM, SIM, SID in the Development of a Global Architecture for Vodafone*. Available from http://www.popkin.com/products/model_agency/vodafonetmforum.pdf . Link checked 1st December 2004.
- [3-30] Kajitani, Nakamura, Hayashi and Miyazzaki. *Migration to Integrated Network Management*. Pacific Telecommunications Council 17th Annual Conference January 22 – 26th 1995, Waikiki, Honolulu.
- [3-31] Martinka, Pruyne, Jain. *Quality of Service Measurements with model-based management for Networked Applications*. Hewlett Packard 2002.
- [3-32] Lewis, Malbon, Pavlou, Stathopolous, Villoldo. *Integrating Service and Network Components for Service Fulfilment*. Proceedings of 10th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 1999. Pages 49 – 62.
- [3-33] Antonio Iera. *Traffic Management Techniques to face the effects of Intrinsic Delays in Geostationary Satellite Networks*. IEEE Transactions on Wireless Communications. Vol 1 No 1, January 2002. Pages 145 – 155.
- [3-34] Enrico Del Re, Laura Pierucci. *Next-Generation Mobile Satellite Networks*. IEEE Communications Magazine, September 2002. Pages 150 – 159.
- [3-35] G Sfikas. *Traffic modelling and Network Management. A Summary of COST 252 Activities*. Undated.
- [3-36] Duresi, Kota, Goyal, Jain, Bharani. *Achieving QoS for TCP traffic in satellite networks with Differentiated Services*. Space Communications 17 (2001). Pages 125 – 136.
- [3-37] Hughes Network Systems. *Spaceway: Network Operations Center Functional Description*, 1999.
- [3-38] L3 Communications. *Graphical Monitor and Control System (GMACS)*. See <http://www.L-3com.com/satellitenetworks>. Link checked 1st December 2004.
- [3-39] *GEMS System Description T5055*. AT & T Ltd, Release 4.2, June 1999.
- [3-40] *SkyWAN Advanced Business Communication by Satellite. Executive Summary, the meshed VSAT-network solution*. Nortel-Dasa Ltd, November 1999.
- [3-41] *LinkWay 2000 Network Management System*. Comsat Laboratories, 1998.

- [3-42] Moshe Zelniker. *Digital Transmissions: NMS*. Satellite Communications Magazine, 1999, Volume 23, No. 7. Pages 38 – 39.
- [3-43] M Stein. *Management and Control of receivers in a satellite distribution network*. SMPTE Journal, 2001, Volume 110, No. 2. Pages 89-93.
- [3-44] Keith Ramsay. *Rural Development Using Advances in Satellite Network Management*. Pacific Telecommunications Council 17th Annual Conference January 22 – 26th 1995, Waikiki, Honolulu. Pages 527 – 530.
- [3-45] V Villagra, J Berrocal, P de Miguel. *An Experience on implementing TMN Interfaces for an OBP satellite network*. From project 900272 partially funded by the CDTI (an agency of the Spanish Industry and Energy Ministry).
- [3-46] Guiseppe Tomasicchio, Alenia Spazio. *The SkyplexNet, a new satellite network with traffic control and distributed network management*. Space Communications, 2000, number Volume 16, No. 4. Pages 227-241.
- [3-47] Intelsat. *Broadband Satellite System Requirements*, enclosure 2. Document BSS-1999-REQ-PRD-005 (From Intelsat). June 1995.
- [3-48] *Value Chain Roles and Relationships*. TeleManagement Forum white paper, Version 0.4, February 2002.
- [3-49] Telemanagement Forum document number GB921. *eTOM, The Business Process Framework for the Information and Communications Services Industry*. Member evaluation version 2.5. Page 35
- [3-50] ETSI BSM working group. *Broadband Satellite Multimedia; IP over Satellite; Standardisation Issues*. TR 101 984, V0.0.8, November 2001.
- [3-51] The European Commission. Directive 1999/5/EC of the European Parliament and of the Council, 9th March 1999. Article 4 paragraph 2. Official Journal of the European Communities L91/10. Available from http://europa.eu.int/eur-lex/pri/en/oj/dat/1999/l_091/l_09119990407en00100028.pdf. Link checked 14 August 2004.
- [3-52] Hughes Network Systems. *Network Operations Control Center (NOCC) Functional Description*. Spaceway North America, 1999.
- [3-53] B Rojat; P deSantis and P Yeung. *Intelsat's Broadband System Studies*. IEE colloquium on Future Broadband Satellite Systems, 1998.
- [3-54] J Wakeling. *Managing Satellite FastPacket networks: A Service Providers perspective*. IEE colloquium on Future Broadband Satellite Systems, 1998.

- [3-55] F Qayle. *Full Services Access Network Requirements Specification*. Issue 3, 1998.
- [3-56] M Fitch and A Fidler. *An overview of Satellite Access Networks*. BT Technology Journal, Volume 18, Number 3, July 2000.

4 Management and measurement of QoS parameters

4.1 Introduction

Following the analysis in chapter 3, which identified QoS parameter management and measurement as being a major difference between terrestrial and satellite networks worthy of further investigation, chapter 4 now proceeds to choose suitable QoS parameters and discuss methods for their management and measurement. This will form a context for later chapters where modelling takes place. Although the primary focus is on satellite links, some information in this chapter is relevant to other types of networks.

In section 4.2, a QoS management process and management ‘modes’ are introduced to give a structure for QoS parameter management. In section 4.3, a summary is given of in-band and out-of-band management traffic and their relationship with active and passive measurement techniques.

Section 4.4 gives a set of assumptions made about QoS parameter measurements pertinent to a satellite link, followed by down-selection of QoS parameters and an overview of the methods of measurement employed by standards bodies such as the ITU and IETF.

Finally section 4.5 provides an overview of the state-of-the-art in QoS parameter measurement and management. This begins with a definition of terminology, then a summary of research activity is given, followed by an appraisal of QoS standards activity and finally a brief survey of measurement tools. The section ends with a discussion about the use of quantiles and percentiles and their relationship with delay probability density functions (pdfs).

4.2 QoS management process and management modes

The concept of a QoS management process is introduced, where a measurement takes place, a decision is made and there is an action to implement the decision. This is very similar to the Hewlett Packard approach mentioned in section 3.7, which separates measurement from diagnosis and management. Such a process is shown in figure 4-1.

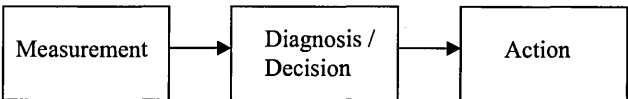


Figure 4-1 Management process

The management process is assumed to be at the network management layer, i.e. above the element management and below the service management (see section 3.4), but the measurement and action functions have implications on the element layer, such as increased overhead traffic. The QoS management role is comparable with other network manager functions such as topology mapping, network resource management and network fault mitigation.

Now the concept of management modes is introduced, based on satellite link congestion state. Three are chosen as shown in table 4-1.

Management mode	Criteria	Actions
Normal	Low network load	None
Intermediate	Medium network load	Increase bandwidth, find alternative routing
Damage limitation	Network load too high	Urgent measures to limit SLA damage

Table 4-1 Preliminary management modes

These modes are re-visited and developed in chapter 6 by quantifying the loads for various traffic types. It also turns out that the QoS measurement method may be changed for certain of the modes.

The actions are what is done to optimise the performance of the network (technically and economically) to satisfy the requests from the service management layer as closely as possible. Examples of such actions for a satellite link might be (1) altering a satellite access scheduler, (2) re-routing around constrictions (3) reporting problems to higher management layers for escalation of action such as fault management, policy management and SLA policing.

It is useful to distinguish between in-band and out-of-band management traffic, as follows. In-band traffic accompanies the user traffic and hence has the same priority in queues as the user traffic. Examples of in-band traffic are probes, trace packets and packet tags. These are generally what are known as ‘active’ measurements, where a real-time overhead is added. Measurements that do not involve adding overhead traffic are known as ‘passive’.

Whereas in-band management traffic accompanies user traffic, out-of-band traffic is transmitted through a network separately from the user traffic and usually carries a high priority in queues. It is generated by the management process by events; examples are timing and fault correlation, alarms and reconfiguration commands.

The management process may use both in-band and out-of-band traffic; for instance in figure 4-1, the measurement process may use trace packets, which is in-band, while the resulting action may require a configuration command to be sent out-of-band.

4.3 Choice of QoS parameters relevant to satellite

There is a large number of QoS parameters, such as throughput, delay, delay variation, packet errors and packet loss; so that it is necessary to select out two or three that are particularly relevant to satellite links for further study.

To aid the selection process, a set of assumptions has been worked out from a mixture of research and practical experience with satellite links.

4.3.1 Assumptions made and rationale

This section contains a number of assumptions made and the reasons for making them, in order to abstract the essential issues that are relevant to satellite links.

Assumption 1: Congestion only occurs on the ground

It was established in chapter 2 that the vast majority of satellites are transparent, i.e. they operate only at layer 1 (physical) of the OSI model. For these satellites, there is no on-

board queuing and hence all the congestion is on the ground. As for the OBP systems, only the broadband ones would have suffered congestion on board, due to uncertainty in the downlink rate. Skyplex is an operational OBP system but congestion does not occur on-board because it is a fully synchronous system, in that the up-link earth stations transmit in sympathy with an on-board clock at a known data rate.

So, it is reasonable to assume that the only place where contention is considered is on the ground, at earth stations where data waits in queues for transmission over the satellite link. Figure 4-2 shows a typical system where three users with PCs are connected to a LAN and then via a VSAT terminal to a gateway earth station via two-way satellite link.

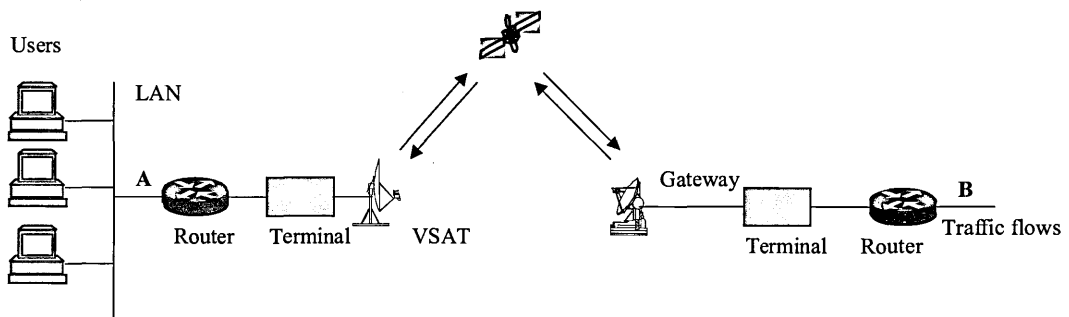


Figure 4-2 Remote users connected by 2-way satellite link

The satellite link is in the access network and the two earth stations transmit to the satellite in their allocated time and frequency bursts allocated by a scheduler at the gateway. Queues will develop at A and B due to traffic bursts, containing packets waiting to be transmitted over the satellite link.

Assumption 2: Contention is not relevant

Separate from congestion, contention is where satellite capacity (or any other capacity) is over-booked. Contention does not mean that more than one user transmits to the satellite during one time-slot, but rather that there are multiple simultaneous requests for bandwidth sent to the scheduler which then allocates the available transmit time-slots to terminals, using a multiple-access technique such as that described in section 2.4.7. If there are no time-slots available, the terminal queues the packets (at least until the time-to-live (TTL) expires). For example, a 2 Mbit/s satellite link may be shared among 50 users that results in a mean of 40 kbit/s each, but it is unlikely that all 50 users will want to transmit at once, so that typically more than 40 kbit/s is available.

The approach adopted by satellite Internet companies like Gilat and Plenexus in marketing this situation is to advertise the 2 Mbit/s to everyone and call it 'maximum' and then impose the 50:1 contention ratio. This kind of contention ratio is fairly common on access networks; for example in the UK, ADSL (over copper) is dimensioned by BT and Tele-West for a contention ratio of 20:1 for corporate users and up to 50:1 for individual homes [4-1].

Contention contributes to congestion and the management process is much the same as for any source of congestion, therefore contention will not be considered as a separate parameter.

Assumption 3: Over the satellite link, either all packets are perfect or all packets are lost

The relationship between bit-error ratio (BER) and received signal-to-noise ratio (SNR) on satellite links is very steep, because M-ary modulation schemes are typically employed (such as QPSK and 8-PSK), coupled with error-correction codes. A 1dB change in SNR can cause several decades of change in BER from the demodulators.

Figure 4-3 shows theoretical plots of BER against SNR expressed as transmitted energy per user bit (E_b) divided by noise density (N_0), taken from Feher, Digital Communications [4-2].

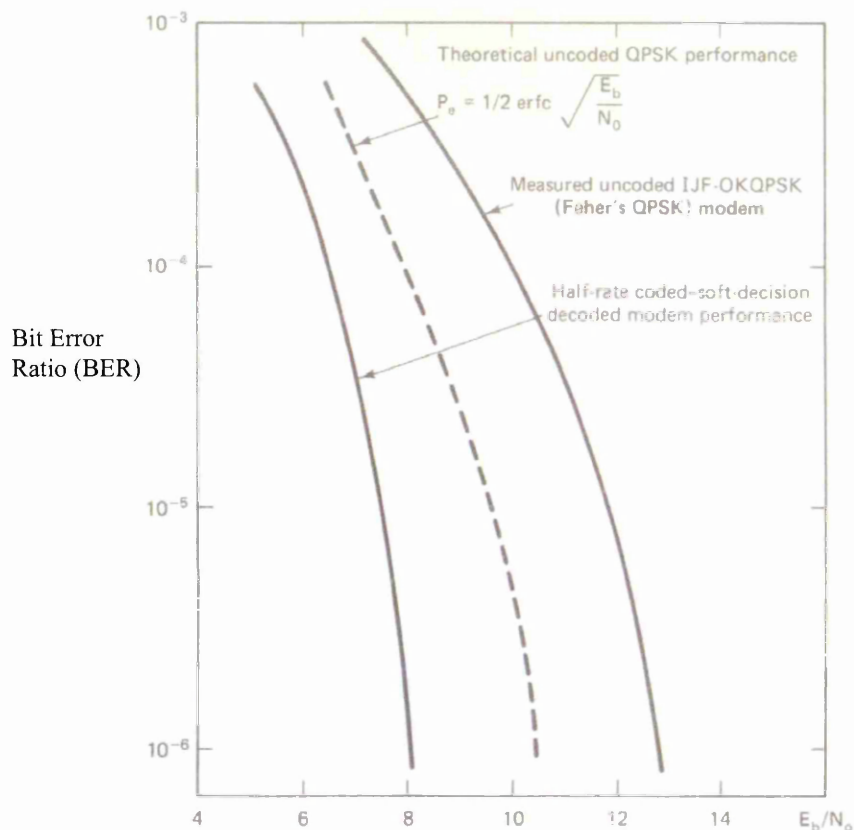


Figure 4-3 Bit-error ratio curves (y-axis) against E_b/N_0

Source: Feher, Digital Communications

Figure 4-3 shows that 1dB change in E_b/N_0 changes the BER by several decades. The curves show the performance of a fairly common modulation scheme (QPSK) with and without an additional half-rate convolutional code. The right-hand plot is from a practical modem, showing that the implementation margin (i.e. the difference between theoretical and practical performance) is around 2dB. The curves would be even steeper if a higher level modulation scheme and / or a more powerful code is used.

So when a satellite link becomes unavailable due to a fade in bad weather, the data it is carrying generally becomes totally corrupted for the period of time that the fade lasts. Once the fade causes the signal to drop below the fade margin, the errors build very quickly due to the steepness of the BER / E_b/N_0 curves and the channel goes quite suddenly from being virtually error-free to total corruption. In this sense, the state of the satellite link is

approximated to binary: it is either on or off. If the link is off, then all packets will be lost for a time. If we have developed a measurement system that will detect a few packets lost in a queue, it should certainly detect packet loss when the satellite link becomes unavailable. If contiguous packets are detected lost over a period of several seconds, then unavailability of the satellite link should be suspected.

4.3.2 QoS parameter selection

A typical list of QoS parameters, related to IP packets on all types of networks, is given in ITU-T recommendations I.380 [4-3] and Y.1541[4-4] as follows:

- delay, measured in seconds
- delay variation, measured in seconds as a percentile or quantile (more on this later)
- packet loss, measured as a percentage of transmitted packets
- throughput, measured in packets/s or bits/s
- errored packets, measured as a percentage of received packets
- spurious packet insertion, measured as a percentage of received packets

Of these, only the first three, delay, delay variation and packet loss are considered peculiar to satellite. The reasons for discounting the others are:

- that the throughput is not a function of the satellite link performance, but rather one of dimensioning and scheduling. Loss of packets in queues will reduce throughput, but this can be measured directly as packet loss rather than its effect on the throughput.
- that errored packets would not occur, since the assumption has been made that either all packets are perfect or all are lost for the duration of a fade
- spurious packet insertion would not occur on satellite links, since the highest they operate in the OSI stack is at layer 2.

Therefore the QoS parameters most relevant to satellite links carrying packet data are:

- Delay,
- Delay variation,
- Packet loss.

It is interesting to relate this set of QoS parameters to typical services.

Table 4-2 shows a set of services and their tolerance to these QoS parameters, from work carried out by the author in support of an article for the BT Technology Journal [4-5]. An estimate of throughput is also given, as is some typical times for two-way satellite link round-trip times (RTT).

The services in table 4-2 are categorised according to layer 4 (transport layer) protocol, either TCP or RTP/UDP; this basically separates the services into non-real-time and real-time respectively. Internet Protocol (IP) is assumed at layer 3 (network layer) since this is very widely used. The tolerance to the QoS parameters of delay, delay variation and packet loss is given in columns 3, 4, and 5. Column 6 is an estimate of the throughput that these services demand.

L 4	Service	Packet delay tolerance	Packet delay variation tolerance	Packet loss tolerance	Throughput Mean / peak (kbits.s ⁻¹)	Max Outage Tolerance		
						LEO RTT 140 ms	MEO RTT 330 ms	GEO RTT 533 ms
TCP	LAN Interconnect	Medium	Medium	Low	1000/10000	55 sec	45 sec	30 sec
	Web Browsing (HTTP1.0/1.1)	Medium	Medium	Medium	10/100			
	File Transfer (e.g. SW backup)	Medium	Medium	Low	10/10000			
	Email	High	Medium	Low	1/50			
	Content Distribution (point to multipoint)	High	High	Low	1000/4000			
RTP/UDP	Voice/Audio Streaming	Low	Low	Low	5-64/5-64	2 sec		
	Video Streaming	High	Medium	Low	1000/1000	2 sec		
	Content Distribution (IP Multicast)	High	Medium	Low	10/64	N/A (e.g. 1 hour)		
	Interactive voice	Low	Low	Low	5-64/5-64	1 sec		

Table 4-2 Some satellite services and required QoS parameters

Column 7 describes the outage tolerance; note that 140ms, 330ms and 533ms refer to round-trip delay times for LEO, MEO and GEO satellites respectively. The outages of 55 seconds, 45 seconds and 30 seconds for TCP are results from simulations of TCP behaviour and give the time taken for the TCP sessions to time out. These simulations were run as part of this study to try and fix an upper bound on the required management action time. These times are discussed further in chapter 6.

This dependency of session time-out upon RTT is significant because, with a GEO satellite, it indicates that TCP sessions will time-out if outages due to fades last more than 30 seconds, even if the session time-out is set to its default value of 1 minute. This is because of Karn’s algorithm, which uses the RTT to determine the start-point for re-transmission intervals. When a TCP link is interrupted, the sender no longer receives acknowledgement (ACK) packets from the receiver, which causes the transmitter to stop sending packets once it has reached the limit of the transmit window. The transmitter keeps a running estimate of the RTT; Karn’s algorithm uses this estimate in an equation that sets the time interval until the first re-transmission attempt of the next packet that the receiver is expecting. This packet is indicated by the latest ACK packet that arrived at the sender. If this fails, then subsequent re-transmit intervals are lengthened further. If no ACK packets are received within the session time-out (typically set to 1 minute), the session is terminated, so that the higher the RTT, the fewer the re-transmission attempts can be made before session time-out.

The figures of outage tolerance for RTP/ UDP services are estimates of what users will tolerate, e.g. a 2 second gap in watching a movie.

Of all the applications listed in table 4-2, the most demanding is interactive voice. Satellite links are generally not very well suited to highly interactive applications such as voice and gaming, because of the long RTTs. With voice in particular, the long delay is a nuisance, not especially because of the response delay, but because of the echo. Telephone instruments commonly have hybrid circuits to convert the 4-wire microphone and earpiece connections to 2-wire lines which, due to their imperfect balancing, cause an echo to be reflected back down the path.

4.3.3 The measurement of delay, delay variation and packet loss

Measurement of delay, delay variation and packet loss is usually performed through the use of measurement points in a network. The measurement points can be located in switches or routers along the path or at terminals. Figure 4-4 shows a generic communications path across a network and two measuring points, MP1 and MP2, with processors attached that are designed to measure delay and packet loss between MP1 and MP2. Since the measurements are made between two points, the method is called '2-point' and has the advantage that measurements made are not affected by delay or losses before MP1.

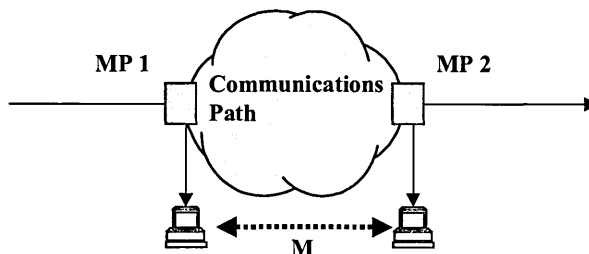


Figure 4-4 Communications path with two measurement points

Attached to MP1 and MP2 in figure 4-4 are processors that perform passive or active measurements. The management process correlates readings taken at MP1 and MP2 to arrive at, for example, per-flow packet count and packet loss. The correlation traffic is represented by M and this traffic is out-of-band, carried at a high priority across the network since its loss would mean loss of valuable information. When conducting active measurements such as delay and delay variation, probes or trace packets could be introduced at one measurement point and recovered at the other.

If there is a satellite link in the path, this link will be the most expensive part of the path and it will have the greatest bandwidth restriction, therefore a minimum amount of management traffic should be carried (both in-band and out-of-band). In cases where the satellite link is delivering broadcast or multi-cast services to many users, there will be several instances of MP2 – perhaps one attached to every remote network that the satellite serves. If the topology is broadcast, M may flow in one direction only, towards the user. So, any measurement system suited to satellite links is required to be (a) scalable to multicast or broadcast topologies without M growing large and (b) tolerant to asymmetric and one-way services.

The 2-point measurement technique has been around for a while; it is specified for measurements of cell transfer performance for ATM by the ITU-T in recommendation I.356, ATM layer cell transfer performance [4-6]. Later it has been specified for measuring

packet transfer performance for IP by the ITU in recommendation I.380 [4-3] and by the IETF in RFC 3393 [4-7].

In fact, although this study focusses upon IP packet transfer, the QoS related standards for IP are generally built upon the rather more mature standards that were developed for ATM; therefore a brief overview of ATM QoS classes is considered worthwhile. Recommendation I.356 defines four ATM cell QoS classes from stringent (class 1) down to Unspecified or 'best effort' (class U). It then goes on to allocate limits to QoS parameters for each class; these parameters include Cell Transfer Delay (CTD), Cell Delay Variation (CDV), and Cell Loss Ratio (CLR).

In I.356, CDV is measured as 2-point and for class 1 the bound is 3ms as the interval between upper and lower 10^{-8} quantiles. The class 1 CTD is 400ms and CLR is $1 * 10^{-7}$. The class 1 CTD is tricky to meet if there is a GEO satellite link delay of 240 – 280 ms in the path as well as other delays due to queues and modems.

Whereas I.356 relates to ATM cell transfer performance, ITU-T recommendation I.380 defines similar parameters for IP packet transfer and performance [4-3]. I.380 includes topologies for MPs and contains diagrams like figure 4-4 for national and international network segments, making use of the 2-point method of measuring packet delay and loss. In I.380, the packet delay variation (PDV) can be quoted as interval based or quantile based.

The IETF has also produced recommendations for IP QoS measurements. RFC 3393 defines measurement methods and parameters for Internet PDV (IPDV) and also suggests that the limits on IPDV be quoted as percentile or quantile [4-7]. RFC 3393 describes several ways of measuring IPDV, based on one-way-delay measurements of pairs of packets that can either be adjacent to each other or separated by agreed amounts. These packets are generated specifically to take measurements and are time-stamped and carry sequence numbers and are therefore 'trace' or 'probe' packets. The RFC is useful in that it gives guidance about design of de-jitter buffers in receivers. There is no guidance given about the length of trace packets, nor on their required intensity. The inter-arrival times of the trace packets are recommended to follow a Poisson distribution to minimise measurement bias and formulas are given for the removal of skew in clocks between the originating and sending networks.

All of the standards recommend specifying transfer delay (IPTD) and packet loss as means, but recommend specifying delay variation (IPDV) as a quantile or percentile on the distribution of the IPTD.

The reason that percentiles and quantiles are recommended by the standards bodies for IPDV is that it helps designers design de-jitter buffers, by making possible such statements as '>95% of packet delay variations should be within the interval – 30ms, + 30ms' (for quantile) or 'the difference between the 99.5%-ile and the 0.5%-ile is <100ms' (for percentile). What is interesting here is that, for such statements to be meaningful, some knowledge of the probability density function (pdf) of the delay is required, the shape of which may be typically heavy-tailed and dependent on network and traffic characteristics. In fact RFC 3393 recommends this method; first construct the pdf, then describe the IPDV in terms of percentiles against the pdf shape.

In general the delay pdf shape will not be known and will have to be estimated. It cannot be assumed to be normal, especially when the traffic pattern is self-similar (more on this in chapters 5 and 6). There are three way of estimating the pdf, (1) by calculation (2) by measurement and (3) by guesswork. Of these (1) is most difficult; the pdf can only be calculated from theory for simple cases, such as for Poisson traffic, making use of techniques such as the Laplace transform; this is explored further in chapter 5. For more practical and complex flows the calculations quickly become intractable and we resort to (2) or (3). Given a suitable measurement method and sufficient time where traffic flows take place without routing changes, histograms of the delay can be plotted and normalised to obtain pdfs.

ITU-T recommendation Y.1541 goes further than I.380 in that it defines metrics for delay, delay variation and packet loss for IP flows across national and international networks [4-4]. It also goes further than RFC 3393 by giving the bound on end-to-end delay variation as $1 * 10^{-3}$ quantile of the ‘underlying’ distribution of the transfer delay in order that manufacturers can ‘assess the necessary size of a de-jitter buffer’. It does not give any clues as to what the ‘underlying’ distribution might be, but it is not generally normal. Various QoS classes are defined that map various times to this quantile. Table 4-3 is an extract from Y.1541 to illustrate their thinking.

		QoS classes			
	Nature of the network performance objective	Class 0	Class 1	Class 2	Class 3
IPTD (IP transfer delay)	Upper bound on the mean	150ms	400ms	1s	U (= best effort)
IPDV (IP delay variation)	Upper bound on the $1 * 10^{-3}$ quantile of the IPTD minus the minimum IPTD	50ms	50ms	1s	U
IPLR (IP Loss Ratio)	Upper bound on the Loss Ratio	$1 * 10^{-4}$	$1 * 10^{-4}$	$1 * 10^{-4}$	U

Table 4-3 Extract from ITU recommendation Y.1541 table 1.

Any network path that includes a GEO satellite link will necessarily be class 1, 2 or 3, since the per-hop delay is 240ms minimum.

4.4 State-of-the-art on QoS parameter measurements and management

It is necessary to report on the current state-of-the-art against which this research makes a contribution in later chapters, but a search shows that there is an overwhelming amount of activity on QoS parameter measurement and management, so that an exhaustive review is not possible in the available time or space. The approach taken here is to focus on the state of activity on satellite QoS plus a few on general QoS that seem to give relevant land-mark results.

4.4.1 Research activity on QoS management

Just as ATM preceded IP in the standards development of QoS measurement methods, so there has been quite some research done on ATM QoS management. For example an algorithm that performs management of on-board queues for ATM cells is described by Iuoras et al [4-8]. They simulated an algorithm called ERICA, which basically organises the load factors in accordance with ATM QoS and sorts the cells into queues of different priority. The QoS is requested from the application using resource management cells. This is an example of much research performed in general on OBP satellite systems despite the fact that very few are likely to succeed in the foreseeable future.

What is interesting about the Iuoras work is the segregation of different priority ATM cells into different queues, an idea that has been transferred to IP packets, e.g. with Diffserv.

There is a body of research into the management process for different priority queues at satellite hubs or earth stations, sometimes coupled with optimisation of schedulers. For instance, the EC 5th framework project SATIP6 has a work package entitled Satellite QoS architecture, which has published a report upon the performance of TCP/IP over DVB-S [4-9]. They have developed a bandwidth on demand (BoD) scheme and suggest separating IP packets into queues with different priorities.

Yet another example of separating IP packets into different queues is by EMS technologies in a Technical Note delivered to the European Space Agency (ESA) study on QoS management over DVB-S and DVB-RCS [4-10]. This ESA study is at the forefront of work on QoS management and is currently underway by Alcatel Space, BT and EMS Technologies. The network management queue (for out-of-band traffic) is the highest priority, with the various traffic classes queued with decreasing priority until the lowest priority (best effort) is reached. This concept of placing packets in different priority queues is picked up later in this chapter in the discussion on diffserv.

Sirisena and Hassan have developed an interesting feed-forward method of minimising variance of queue lengths and tested it with Poisson and self-similar traffic, with various 'greedy' traffic sources combined with high priority traffic sources. Their research field is adjacent to this and they conclude that Poisson traffic is the worst case from their point of view because it is memoryless and therefore cannot be predicted, so that self-similar traffic with its long-range dependence benefits to a greater degree from their algorithm [4-11].

Whereas their result is correct, i.e. self-similar traffic does benefit more than Poisson from their algorithm, it is shown in chapter 6 that the reason is not that Poisson is memoryless, but that self-similar traffic generates much more variance in queues for the same load and so shows a greater reduction. Their algorithm uses a sampling method whereby the sampling interval is greater than the round-trip time. This approach was probably taken because time delays tend to make control systems unstable, and this tendency to instability is avoided in their paper by letting the system settle between samples. The difficulty that this causes in the case of a GEO satellite link is that round-trip times can be 600ms or more and, as the response time has to be of the order of a few seconds, this allows only a very small number of samples to be analysed before decisions are made. For this reason, their algorithm is not suitable for satellite links. However on the positive side, their work shows that controlling the load on satellite earth station queues is an alternative to limiting their lengths and it will tend to (desirably) push the congestion out towards the edges of the network, i.e. away from the satellite link. They also confirm

the findings in chapter 6 that self-similar traffic in unlimited queues with no control leads to very large variations in delay.

Chai and Sun have taken the approach where traffic is segregated into elastic and inelastic classes, where elastic traffic is tolerant to changes in delay and throughput without affecting the QoS of the application [4-12]. In this case, TCP or application-level reliability recovers lost data. Inelastic traffic is not tolerant and examples of such traffic are from streaming applications. The findings in this paper compare favourably with the information in table 4-2. The paper goes a little further by proposing some figures for BER and delay variation that can be tolerated by user requirements. Now there is a practical difficulty with specifying BER in this context, in that it can never be measured on practical links because it is not stable enough to make measurements. The usefulness of BER is restricted to calculation of link budgets. Nevertheless the paper describes simulations of measured traffic against background traffic and maps bit-errors into packet errors. The paper does not address how packet delay may be measured in a practical network. The most relevant result in their paper, which agrees with the findings in this study, is that the delay pdf is long-tailed and is sensitive to link characteristics.

Moving now from queue management to in-band management traffic, one of the interests in this study is to ascertain and minimise the in-band management traffic intensity since satellite capacity is expensive. This also has been studied in the ATM case and the ideas transferred to IP packets.

Measurements of management traffic on networks of ATM switches have been made by Heo et al [4-13]. They use Jackson's theorem to calculate management traffic intensity that results from call processing on large networks with many nodes. They use TMN as the architecture for the commands. They have performed simulations of large networks using Poisson arrivals and compared their results with the theory. Jackson's theorem is a way of multiplying queue characteristics over a large network to arrive at overall queue characteristics so, for example it proves that a network of queues can be modelled as one queue with a load that is a multiple of the individual queue loads. However, Jackson's theorem it is not used in this study since it assumes that the arrivals into the network are Poisson distributed, which is not the case for Internet traffic. Further, Jackson's theorem assumes that the service times are exponentially distributed, which will not be the case for packet networks (more on this in chapter 5).

Sampling is a common technique used to estimate QoS parameters. This has two main advantages, especially attractive for satellite use:

- to keep the in-band and out-of-band management traffic to a minimum and
- to keep the processing power at the measurement points to a minimum, especially if these are at the remote ends of access links on small user networks.

Research on packet sampling has been carried out by several workers and it is interesting to compare their approaches with the one taken here. One critical issue is the relationship between sample size and accuracy of the estimation of QoS parameters and Cozzani has done some good work forming this relationship for ATM cells [4-14]. For Poisson traffic, she shows that a sample size of about 40 is necessary to estimate a cell loss ratio of 0.125 to within 10%, by simulation and from theory. Cozzani starts with the theory of large deviations and classical sampling theory to assess the minimum set of samples necessary for a given accuracy of mean delay and loss measurements to the ATM cells. Estimation of

delay variation is not attempted. Only Poisson traffic is tackled in this paper, since it is much better behaved than self-similar traffic as will be pointed out later in chapters 5 and 6.

A relationship between sample size and estimation accuracy is developed by Zseby [4-15], for IP packets. The maximum sample intensity is set at 5%, which results in only 1.5% of the estimations of delay mean and variance falling within 1% of the true value, using random sampling. Zseby shows that changing the sampling strategy from random to systematic and then to stratified sampling increases the percentage of estimations that are within 1% accuracy up to a maximum of nearly 94%. This looks very good for stratified sampling; this is because the process of stratification necessarily reduces the variance being estimated, since it divides the population into 'strata' of values and samples each independently. This effective reduction in population variance will cause the estimations to become more accurate, because confidence interval equations have the population variance in the numerator; chapter 6 explains this with some example calculations. A useful outcome from Zsebys work is identification of the problem of synchronisation between measurement points, that this requires management overhead traffic but unfortunately he or she does not offer any results on this. The sampling method employed requires substantial processing power at the measurement points to capture and decode packets and to add ID tags. The paper discusses size of ID tags and use sizes between 16 and 320 bits (2 – 40 bytes) according to where in the IP packet header the tag is placed and how many IDs are needed.

Claffy and Polyzos [4-16] also compare random, systematic and stratified sampling to estimate IP packet delay and also include a comparison of packet-triggered sampling with time-triggered in the systematic and stratified cases. In order to compare the various sampling techniques, they use goodness-of-fit metrics which divides up the measurement scale into bins, where the expected quantity in each bin is derived from the delay pdf. They show that systematic and stratified time-driven sampling methods give a better score on the goodness of fit, with stratified giving slightly better results than systematic; at least this is shown for sampling intensities greater than about 1%. Below this the results are so noisy it is difficult to decide.

Another angle on sampling is provided by the IETF; Poisson distribution of intervals between pairs of sampled packets is recommended in RFC 3393 when measuring IP packet delay variation in order to avoid false results from aliasing [4-7].

Drobisz [4-17] comments that sampling reduces required processor time and estimates that, without sampling, 300 MIPS is needed to analyse traffic from his Gigabit Ethernet carrying 64 byte packets. He describes an adaptive sampling algorithm that protects the CPU. He compares the performance of his adaptive sampling scheme with static sampling for random and stratified sampling, arguing that the adaptive technique gives more accurate results. For stratified sampling he gives percentage errors on mean and variance of packet counts as roughly 10% when sampling one packet in 50 (2%).

Some relevant results from Zseby and Drobisz are brought together in table 4-4.

	Sampling intensity	Accuracy
Zseby	5% stratified	94% of sample values within 1% of true delay
Drobisz	2% stratified	Within 10% of true values of delay mean and variance

Table 4-4 Some results of estimation accuracy against sample intensity

The figures in table 4-4 are commented on in chapter 6 where QoS measurement methods are compared. Stratification improves accuracy by dividing up the population to reduce the variance and is a popular sampling technique but it demands a high degree of processing power. Several workers have commented on processor power placing the limit on the rate at which OSS events and measurements can be handled. On a satellite link, stratification is not attractive since the available processor power at remote networks is likely to be small.

Systematic sampling is used in this study, i.e. on a constant time grid, the reason is to minimise the processing power required at the remote network. Poisson sampling was tried in a few of the simulations in this study as an alternative to systematic sampling and found to make no difference.

Phaal also discusses the concepts of packet sampling and gives some useful formulas for relating sample mean to population mean, his confidence interval formulas assume that a large number of samples is taken so that the estimation error distribution approaches normal by the central limit theorem [4-18]. This is a necessary condition for calculating confidence intervals in classical sampling theory, also adopted by this study in chapter 6.

The size of the management traffic to support correlation of samples is not given in any of the above papers, but it can be shown that overhead traffic can equal the user traffic if a measurement method like Zsebys is used, bearing in mind that packets on the Internet are generally between 40 and 1500 bytes according to measurements performed by Claffy on the MCI network [4-19]. Such correlation would need to be achieved using a protocol like SNMP. Two examples of SNMP overhead traffic intensity are given to show that this would incur a large amount of management traffic overhead. The first is taken from work by the author (on another project) and indicates that, to recover single MIB values using Get and Get Response commands, 350 bytes have to flow in each direction [4-20]. The second is taken from a study on SNMP by Pattinson [4-21], which found that SNMP traffic flows are around 100 – 250 bytes for health checks and fault-finding messages.

Moving now to IP QoS management mechanisms, the IETF working group ‘Intserv’ has, according to their web-site, the remit to develop the necessary interfaces and requirements for an Internet service model to support integrated services, i.e. audio, video, real-time and data services. Intserv is now largely legacy and defines the concept of ‘flow’ as a stream of packets with the same destination address, source address and port number. It requires routers to maintain state information on each flow to determine what flows get what resources based on available capacity. Within Intserv, the Resource reSerVation Protocol (RSVP) is a popular QoS signalling protocol used by hosts to request network resource with a specific QoS. The problems with Intserv are scalability (since every router must contain state information on every flow) and lack of policy control mechanisms.

Diffserv overcomes these problems by aggregating flows and assigning classes to them for appropriate treatment in a network. It defines a set of per-hop behaviours and traffic conditioners / shapers. Diffserv processes include admission control where the network can refuse traffic, packet scheduling and sub-streams usually implemented using queues of different priorities. Figure 4-5 shows an example of a queuing process within a diffserv router.

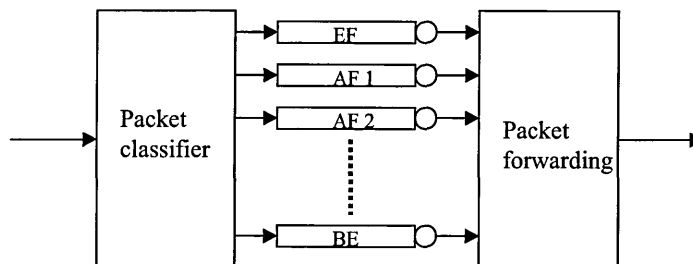


Figure 4-5 Example of priority queuing

The packet classifier decodes the header information type of service (TOS) field and classifies the packets according to a management policy. The packets are then queued according to priority, the highest priority in figure 4-5 is expedited forwarding (EF), then various algorithmic forwarding (AF) classes with traffic management algorithms right down to best effort (BE) which is not managed at all. This prioritisation can be mapped onto OSS traffic; alarms and other urgent event information may well use the EF class, whereas user traffic, non-urgent OSS traffic may use one of the AF classes and non-urgent traffic can use BE class.

Both Intserv and Diffserv working groups have developed MIBs for QoS management [4-22], [4-23]. Huston has made measurements of IP performance using these MIBs and suggests the use of SNMP polling for flow measurements from routers; his paper describes tools for reading the headers of IP packets as they pass measurement points [4-24] much as in figure 4-4. Like Drobisz, Huston comments that a large amount of processor power is required to perform the reading of IP headers. This makes the method not particularly suited to satellite access links since the processor at the remote end is not likely to be large and not likely to be in the management domain of the service provider (see section 3.8.4).

The ‘networking’ working group of the IETF have developed an architecture to allow Intserv services to be carried over Diffserv networks described in RFC 2998 [4-25]. This combines the advantages of both, e.g. RSVP within Intserv in the access network per flow and then aggregation of the flows over Intserv in the core network. The interworking of intserv and diffserv aligns well with the migrating use of satellite towards the user (see chapter 2). This is because the use of RSVP is a receiver oriented protocol that forms distribution trees similar to multicast groups, which are naturally suited to satellite services.

Iera [4-26] picks up this thread and gives a framework for integrating Intserv over the satellite access network with a Diffserv core network via an edge router, using Euroskyway as a case study. He performs some simulations of connection set-up times and concludes that interworking is feasible with QoS parameters suited to all but the most interactive of

services (such as voice and games). He makes the point that deployment of effective interworking between terrestrial and satellite segments is difficult because, as he says:

‘satellite bandwidth is a precious resource and propagation delay strongly influences any design decisions’.

There is an increasing use of agents in networks to perform QoS management. For example, the Common Open Policy Server (COPS) is a QoS policy manager and performs operations such as out-sourcing QoS requests from network elements, auto-discovery, queuing and scheduling. It integrates with RSVP (and SIP) and is applicable to Diffserv networks. COPS distributes end-to-end QoS requirements among the Diffserv network segments using a client-server architecture and updates the requirements when Service Level Agreements (SLA) are updated. COPS-RSVP messages travel between routers to facilitate dynamic admission control decisions. For more information on COPS, see [4-27]. Daoud postulates that a mobile QoS agent could reside on an OBP satellite to reduce latency and call set-up times when such a satellite system is integrated with a terrestrial mobile network [4-28].

According to work done on QoS measurements by Corral et al on the Saturne project, one-way delay measurements are becoming increasingly used on the Internet because symmetry of routes cannot be guaranteed [4-29]. The development of one-way delay techniques and metrics is particularly interesting for satellite links, since they are naturally suited to one-way and asymmetric multicast and broadcast applications. SLAs for satellite links often require policing and guarantees for one-way delay.

Asaka reports on a method of traffic flow estimation, including assessment of self-similarity, using SNMP polling [4-30]. Using a 10 second polling interval to update router MIBs, he estimated the flow rate and Hurst parameter H and compared the estimations with actual measurements. This method, although interesting and passive in the sense that it does not interfere with user packet flows, has the drawback that it polls routers; this could add substantially to the management overhead traffic over a satellite link, especially if there are many routers on the remote network side.

4.4.2 Survey of tools for measuring packet delay and loss

There are several off-the-shelf tools available for measuring packet delay and loss that use both passive and active methods and a few are mentioned here to give an idea of state-of-the-art. A more comprehensive list of Internet analysis tools is available from the Co-operative Association for Internet Data Analysis (CAIDA) [4-31].

Another web-site listing network monitor tools is MonitorTools.com [4-32] which lists around 30 tools such as TracePlus / Ethernet that displays statistics by node, protocol and address pair.

TCPdump [4-33] is probably the best known network sniffer and can watch and diagnose network traffic according to rules and write header information to files. It was originally designed for use with Unix platforms, Windump is the porting to the Windows platform of tcpdump [4-34]. Both tcpdump and Windump are freely downloadable from the Internet.

Nprobe, is a passive measurement system that uses large processors to full-rate capture packets from a 1 Gbit/s network to analyse usage patterns including object sizes and persistent HTTP¹ sessions, developed by Deri [4-35].

The Communications Management (CM) Toolset developed during the EC IST project ACQUILA uses active and passive methods to probe and monitor QoS parameters including delay and loss in distributed Intserv networks [4-36]. The toolset includes active probing, router monitoring, packet loss measurements and network load estimation and captures the results in a form suitable for SLA monitoring.

Hofmann et al took the basic CM toolset and enhanced it by fitting GPS synchronisation and an integrated database to improve the accuracy of one-way delay and delay variation [4-37]. They studied the effects on TCP segment delays when a mixture of TCP and UDP are transmitted over a congested link and constructed some histograms of the delay. These do not correlate well with the findings of this research since Hofmann's histograms were plotted from transport layer segments and hence are affected by transport layer functions such as TCP slow-start. The delay histograms in the Hofmann paper appear to be approximately normally distributed whereas the ones in this research have heavy tails. As further work it would be interesting to assess the effect of UDP and TCP mechanisms on delay histograms and to see if they do indeed become more normal.

Pathload is a tool that inserts probes into the network path to assess the amount of unused capacity. It works by measuring the delay and loss characteristics on the probe when it is received at the far end of the path and if there is little delay or loss then it reports that some capacity is available [4-38]. This is a very intrusive tool which is suited to high capacity networks, it is not suited to satellite links or for lower capacity networks since it is likely that the user traffic will be significantly affected by the probes.

eProbe [4-39] is another active QoS measurement tool that launches probes into networks and reports its findings on delay and loss characteristics to agents distributed around the network. A distributed management system then gathers the results and performs trend analysis. But there are no results to show how intense the traffic needs to be from these probes to make effective measurements.

PathChar and its derivative pchar were developed by the University of Berkeley and Sandia National Laboratory respectively, to measure IP packet loss and delay characteristics [4-40]. They both use ICMP responses and vary the time-to-live TTL to probe into networks to see how far packets can travel and this reveals information about links traversed. Multiple repetitions give information about queues and loss. This is another highly active measurement system and a weakness is that it assumes path symmetry in terms of delay and bandwidth, which disqualifies it for the majority of paths that include a satellite link.

Ethereal is a protocol analyser that takes data captured from a network by a tool such as tcpdump [4-41]. It can dissect hundreds of protocols from the link layer to the application layer, such as AAL (link), ARP, H261, ICMP (layer 3), TCP / UDP (layer 4) and FTP, HTTP, COPS (application layer).

¹ Persistent HTTP is where a single HTTP session is used to download several objects, an improvement over the original HTTP that used a different HTTP request (and hence a different TCP session) for each object.

Finally, Corral describes the Saturne platform, already mentioned above in the context of sampling. It is designed to make one-way end-to-end active delay and loss measurements in accordance with the IETF Internet Protocol Performance Metrics (IPPM) working group [4-29]. The method is to insert timestamps in the form of trace packets and recover them later in the path, using GPS for time correlation. RFC 2679 [4-42] and RFC 2680 [4-43] give metrics for packet delay and packet loss respectively. The emission pattern for trace packets in this scheme is variable. Of all of the tools described, this uses a measurement method that is most suited to satellite links, however the authors have not applied it to satellite.

4.4.3 Use of trace packets

In this study it is proposed to adopt the use of trace packets to probe the chosen QoS parameters of delay, delay variation and packet losses on a network. The reasons for taking this approach are that:

- Active measurements using sampling or probing have been established as most suitable for paths that contain a satellite link, provided they do not add significantly to traffic load - and hence do not add significantly to in-band management traffic,
- Trace packets are assumed to carry time-stamps and sequence numbers so that loss and delay characteristics can be calculated at receiving networks without additional correlation traffic - and therefore do not add significantly to out-of-band management traffic,
- Trace packets are used to probe the path, rather than adding tags to user packets, since they can be easily recovered at one or more remote network locations - and therefore do not put large loads onto remote network processors.

With the trace packet method, there are two types of management traffic, one is the trace packets themselves (in-band) and the other is the information that must be recovered from the trace packet processor (out-of-band). Figure 4-6 illustrates the principle.

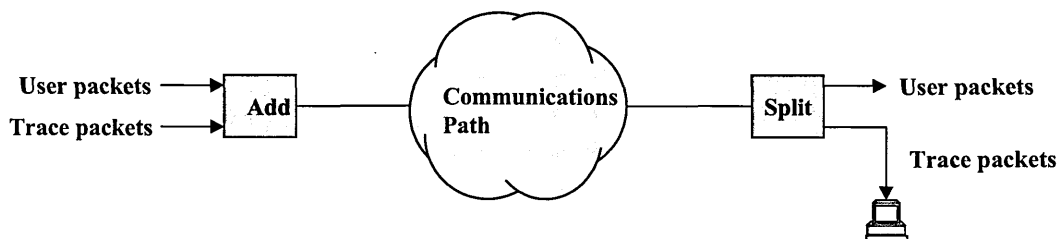


Figure 4-6 The trace packet method of measurement

The trace packet method is adopted and further investigated in chapters 5 and 6 by modelling, simulation and measurements using a test network.

4.5 Summary of chapter 4

This chapter has proposed a choice of QoS parameters to measure and manage that are pertinent to satellite links. These are packet delay, delay variation and packet loss. It has described a management process framework and a measurement method for these parameters, in the context of current state of the art.

The standards recommend that the delay variation is measured in terms of percentiles or quantiles and that measurements of delay mean, delay variation and packet loss are taken

once per minute. One method of measuring percentiles or quantiles is to measure the mean and variance of the delay, for example by using trace packets and then calculate the percentiles or quantiles from a knowledge of the pdf shape, again derived from the use of trace packets.

Chapters 5 and 6 now drill deeper into the modelling and measurement of packet delay, delay variation and packet loss, using trace packets as estimators.

4.6 References for chapter 4

- [4-1] *BT Strategy in DSL market*. Foresight consulting report. Undated
- [4-2] Kamilo Feher. *Advanced Digital Communications*. ISBN 0-13-011198-8 Prentice-Hall 1987. Page 369
- [4-3] ITU-T. *Internet Protocol Communication Service, Internet Protocol data communications service – IP packet transfer and availability performance parameters*. Recommendation I.380, December 2002.
- [4-4] ITU-T. *Network Performance Objectives for IP-based Services*. ITU-T Recommendation Y.1541, May 2002
- [4-5] M Fitch. *The use of Satellite for Multimedia Communications*. BT Technology Journal, Volume 21, Number 3, July 2003. Pages 90 – 101.
- [4-6] ITU-T. *B-ISDN ATM layer cell transfer performance*. Recommendation I.356, version 10/96 October 1996
- [4-7] C Demichelis. *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)*. RFC 3393, IETF, November 2002.
- [4-8] A Iuoras, P Takats, C Black, R DiGirolamo, E A Wilbowo, J Lambadaris and M Devetsikiotis. *Quality of service-oriented protocols for resource management in packet-switched networks*. International Journal of Satellite Communications, Vol 17, March- June 1999, pages 129 – 141.
- [4-9] T Inzerilli and Antonio Pietrabissa. *Satellite QoS Architecture in the SATIP project*. Universita di Roma, Dipartimento di Informatica e Sistemistica. Undated, but known to be 2002 or later.
- [4-10] *Integrated Resources and QoS Management in DVB-RCS networks*. Technical note 2 delivered to the European Space Agency Contract 17224/03/NL/ND. ESA ARTES 1 study ESA4223. Alcatel Space and EMS Technologies, January 04, Version 1.0, page 75.
- [4-11] H Sirisena and M Hassan. *Generalised minimum variance control of queues in packet switching networks*. Communications in Information and Systems, Volume 2, Number 4, pages 385 – 398, December 2002.
- [4-12] WK Chai and Z Sun. *QoS Measurements of Multimedia Traffic via multi-service Satellite System*. Centre for Communications Systems Research, University of Surrey, Guildford. Undated but known to be during 2003.
- [4-13] Dong-Hyun Heo, Sang-Wook Chung and Gil-Haeing Lee. *The Effects of Management Traffic on the Local Call Processing Performance of ATM Switches Using Queue Network Models and Jackson's Theorem*. ETRI Journal, Volume 25 Number 1, February 2003. Pages 34 – 40.

- [4-14] I Cozzani and S Giordano. *Traffic sampling methods for end-to-end QoS evaluation in large heterogeneous networks*. Computer Networks and ISDN Systems 30 (1998), pages 1697 – 1706.
- [4-15] T Zseby. *Deployment of Sampling Methods for SLA Validation with Non-Intrusive Measurements*. Fraunhofer Institute for Open Communications Systems (FOKUS). Undated.
- [4-16] K Claffy and G Polyzos. *Application of sampling methodologies to Network Traffic Characterisation*. Computer systems laboratory, University of California. Not dated, but known to be after 1993.
- [4-17] J Drobisz and K. Christensen. *Adaptive Sampling Methods to Determine Network Traffic Statistics including the Hurst Parameter*. IEEE, 1998.
- [4-18] P Phaal and S Panchen. *Packet Sampling Basics*. Available from www.sflow.org. Link checked 2nd December 2004.
- [4-19] K Claffy. *The nature of the beast, recent Traffic Measurements from an Internet Backbone*. CAIDA, April 1998.
- [4-20] A Fidler and M Fitch. *EC 3rd framework project VANTAGE trial 2 report*. British Telecommunications plc report number SSTR141, August 1997, page 37.
- [4-21] Colin Pattinson. *A Study of the behaviour of the Simple Network Management Protocol*. School of Computing, Leeds Metropolitan University. Undated.
- [4-22] F Baker, J Czerwinski, A Sastry. *Integrated Services Management Information Base using SMIv2 (RFC2213)*, Network Working group of the IETF, September 1997.
- [4-23] F Baker, K Chan, A Smith. *Management Information Base for the Differentiated Services Architecture (RFC3289)*, Network Working group of the IETF, May 2002.
- [4-24] IG Huston. *Measuring IP network performance*. Cisco IPJ Volume 6, Issue 1, March 2003. (Available from Cisco web-site)
- [4-25] Y. Burnett et al. *A Framework for Integrated Services Operation over Diffserv Networks*. RFC 2998, IETF, November 2000.
- [4-26] Iera, Molinaro. *Designing the Interworking of Terrestrial and Satellite IP-Based Networks*. IEEE Communications Magazine, February 2002. Pages 136 – 144.
- [4-27] For information on COPS products, see <http://www.reveal.co.uk/Products/Eion/Cops.htm>. Link checked 2nd December 04
- [4-28] Fawzi Daoud. *Hybrid satellite / terrestrial integration*. Computer Networks 34 (2000). Pages 781-797.
- [4-29] J Corral, G Texier, Laurent Toutain. *End-to-end active measurement architecture in IP networks (Saturne)*. ENST Bretagne. Undated but known to be 2002 or later.

- [4-30] T Asaka, *Method of estimating short-interval traffic distributions using MIBs*. *IEICE Transactions on Communications*. Vol. E85-B No. 5, May 2002. Pages 1038 – 1041.
- [4-31] See www.caida.org/tools/ Link checked 2nd December 2004.
- [4-32] See http://www.monitortools.com/cat_traffic/ Link checked 2nd December 2004.
- [4-33] See <http://tcpdump.org>. Link checked 2nd December 2004
- [4-34] See <http://www.windump.polito.it/> Link checked 2nd December 2004
- [4-35] L Deri. *NProbe: An Open-source Netflow probe for Gigabit networks*. NETikos SpA, Pisa Italy, 2004.
- [4-36] AQUILA Adaptive Resource Control for QoS *Using an IP-based Layered Architecture*. EC IST 5th framework project . See <http://www.ist-aquila.org>. Link checked 2nd December 2004.
- [4-37] U Hofmann, T Pfieffenger and B Hechenleitner. *One-way-delay measurements with CM toolset*. University of Salzburg. Not dated, but known to be 1999 or later.
- [4-38] M Jain and C Dovrolis. *Pathload: a measurement tool for end-to-end available bandwidth*. *Computer and Information sciences*, University of Delaware. Undated but known to be later than 2000.
- [4-39] *ePROBE. The Problem: Distributed Network Management is Difficult and Expensive*. June 2002. Available from http://www.tavve.com/pr_snmp.html Link checked 2nd December 2004.
- [4-40] B Mah. *Pchar: Child of Pathchar*. Security and networking Research department, Sandia National Laboratories, California. July 1999.
- [4-41] For more information on Ethereal, see <http://www.ethereal.com/introduction.html>. Link checked 27th December 2004
- [4-42] *A one-way delay metric for IPPM*. RFC 2679 from Network Working Group of the IETF. September 1999.
- [4-43] *A one-way packet loss metric for IPPM*. RFC 2680 from Network Working Group of the IETF. September 1999.

5 Description of models and test network

5.1 Introduction

This chapter contains descriptions of the simulation models, mathematical models and a practical test network used in this study. It contains calibration results for the models, but not the experimental results; these are given in chapter 6.

The simulation model used in this study is built using the OPNET tool, since it was available and some experience of it had already been gained. The only alternative considered was Block Oriented Network Simulator (BONeS) from Comdisco, but this does not have the on-line help, workshops and discussion groups that are available within the OPNET community. Also, BONeS requires a Unix workstation, whereas OPNET was run on a laptop computer running Windows 2000.

Straight-forward mathematical models are used, largely distilled from literature, in order to calibrate the simulation models. The mathematical description of the traffic and queue models and their implementation and calibration in OPNET are given together, inter-twined so that the relationship of one to the other is easy to see.

A test network is used in this study because it gives valuable information on the usefulness of trace packets on a practical (i.e. non-ideal) network and also sheds light on the limitations of models to describe what happens on a real network.

This chapter continues by giving an overview of OPNET (section 5.2), a description of the traffic models used (section 5.3), a description of the queuing model used (section 5.4) and a description of the test network (section 5.5). Section 5.6 is a brief summary of the chapter.

5.2 Overview of OPNET

OPNET is a communications network design and protocol simulation tool developed by OPNET Technologies [5-1], which has become an industry standard modelling tool. OPNET is an event-driven simulation tool where process remain in a constant state until interrupted, for instance by the creation of a packet or a queue time expiring. Events are generated through interrupt scheduling and the type of interrupt along with an appropriate index determines the response of a process to a particular event. OPNET comes with a library of standard protocols like FTP and TCP and a set of mathematical functions such as statistical distributions that have proved useful in this study. The results from OPNET simulations is usually in the form of graphs such as throughput against time, it does not provide advanced post-processing such as histograms or spectral analysis. When this is required, OPNET will dump raw data to a text file that can then be imported into a program like Microsoft Excel or Matlab.

OPNET models are built up using a three-level hierarchy consisting of the network level, the node level and the process level. An example of a network level design is shown in figure 5-1.

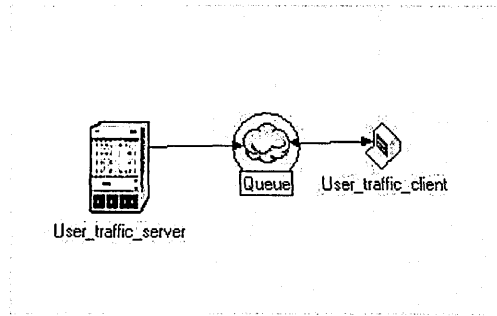


Figure 5-1 OPNET network level design example

Each of the nodes in figure 5-1 has a corresponding node level design, the second level in the design hierarchy. The node level represents the functional elements of a communications node, consisting of packet generators, queues, sinks etc. Packet streams connect the nodes together, defining the flow of data from one node to another. Statistic wires are used to transfer discrete statistics between nodes. Figure 5-2 shows the node level design for the Queue shown in figure 5-1.

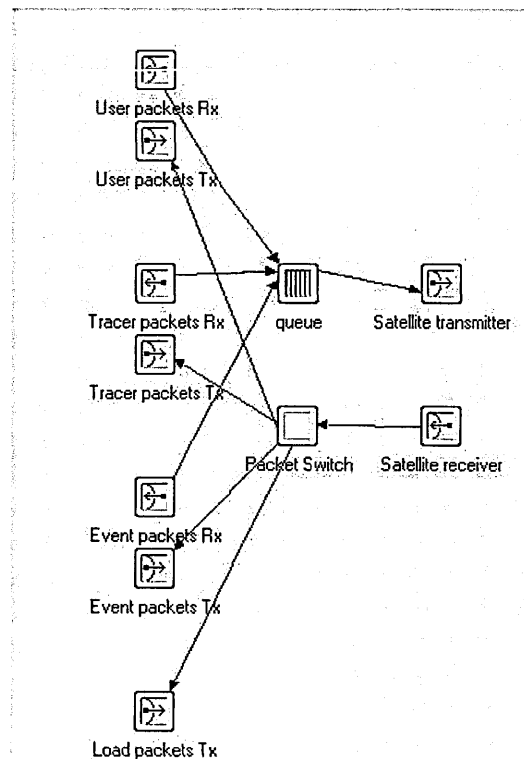


Figure 5-2 OPNET node level design example

The lowest level in the design hierarchy is the process level. A process model is included in every node and is where the detailed protocol mechanisms are designed and developed. Process models are defined using a language called Proto-C, which is an extension of the C

programming language. Proto-C designs consist of a number of states (similar to those found in finite state machines) connected together using transition wires, set up graphically. Only one state is active at any particular time and transitions from one state to another occur as a result of interrupts arriving at the process. The specific state transition followed can be controlled by conditions assigned to that transition.

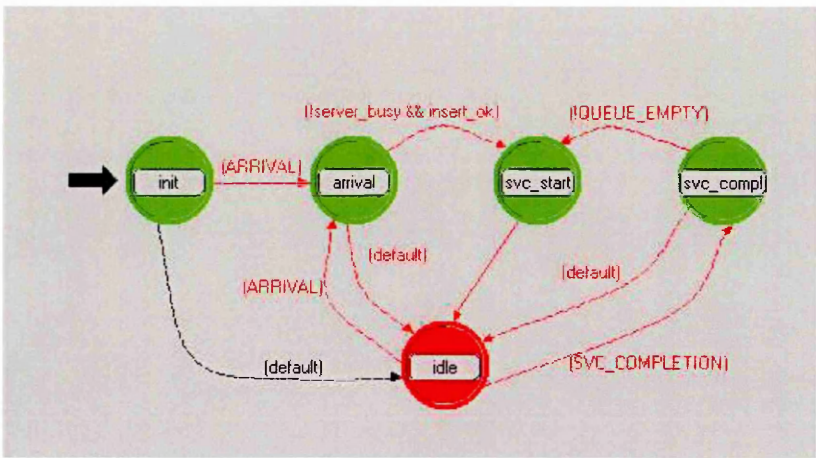


Figure 5-3 OPNET process model design example

There are two types of state in OPNET, forced (green in figure 5-3) and unforced (red in figure 5-3). Each state has associated with it fragments of C-code that are executed when either that state is entered or when exited, these fragments are called ‘executives’. The entry and exit executives of a forced state are executed sequentially with no pause in between, they are usually used to compartmentalise the code into smaller, more manageable blocks. On entry to an unforced state, the entry executive of that state is executed and the process stops, handing back the simulation control to the kernel. At the next interrupt to that process, the exit executives of that state are executed and the transition out of that state to the next is determined by the conditional transition wires based on the type of interrupt. At the start of a simulation, every process model enters and executes an ‘Init’ state which initialises the process with parameters such as simulation attributes and registration of statistics to be collected. Each process model operates in parallel with all the others, which allows simulations of real networks in which each processor in a node operates independently. The parallel behaviour is achieved by time-stamping every event with the instantaneous simulation time so that all events with the same simulation time can be taken into account.

OPNET has a variety of probes for collecting statistics. For example, probes can be assigned to statistics from links for collecting throughput data or to sinks for collecting end-to-end delay data. Probes can collect raw data or pre-process it by, for example, collecting values in buckets and dividing the bucket contents by time intervals in order to provide a running average.

OPNET is a very powerful and flexible tool that requires a degree of expertise in C programming. Two training courses (introduction and advanced) were attended to gain

knowledge and experience with OPNET as part of this study. The code developed for this study is included in Appendix A.

5.3 Traffic models

Two traffic sources are now described, one is the classical Poisson model and the other is a self-similar model, since this has been shown by several workers to more accurately model packet flows on the Internet, see for example Leland [5-2]. The Poisson model is included because mathematically it is well behaved and serves as a useful calibration tool. The fundamental difference between the two is that, with the Poisson model, the probability of a packet being generated in a given time interval t does not depend on the packets generated in previous intervals, whereas the self-similar model has a level of dependence on the past that is determined by the degree of self-similarity. The degree of self-similarity is described by the Hurst parameter H , which has a value between 0.5 and 1, where $H = 0.5$ represents zero self-similarity and $H = 1$ represents exact self-similarity. A Poisson source has no self-similarity so $H = 0.5$ as demonstrated graphically later in this section.

5.3.1 A Poisson model

The Poisson distribution [5-3] can be written as:

$$P_k(t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \quad (5-1)$$

where $P_k(t)$ = the probability that k packets will be generated in time t
 λ = mean packet generation rate.

Equation (5-1) can also be derived as an off-shoot from queuing theory, this is shown in Appendix B. From equation (5-1), the probability of zero packets being generated during a time t is found by putting $k = 0$,

$$P_0(t) = e^{-\lambda t}$$

Let the variable τ be the time between adjacent packets, then the probability that $[\tau > t]$ is $P_0(t)$, so that the probability distribution function (PDF) $A(t)$ that $\tau \leq t$ is

$$A(t) = 1 - P_0(t) = 1 - e^{-\lambda t} \quad t \geq 0$$

And the probability density function (pdf) $a(t)$ is obtained by differentiation

$$a(t) = \lambda e^{-\lambda t} \quad t \geq 0 \quad (5-2)$$

Equation (5-2) is the exponential distribution. Therefore, if a packet generator is required to produce packets according to a Poisson distribution with mean arrival rate λ , it can achieve this by producing packets with inter-arrival times that are exponentially distributed with mean $= 1/\lambda$. This technique was implemented in OPNET using a process based on the Simple Packet Generator library module. The generator is initialised with an exponential inter-arrival time

distribution with a mean ($= 1/\lambda$) that is set at simulation time and produces packets to a format called 'mrf_user' that was created using the packet format editor. This is a simple packet consisting of a 20 byte header and a variable length payload to represent IP packets. Further details of how the Poisson packet generator was implemented in OPNET are given in Appendix A.

5.3.2 A self-similar model

There are several ways of modelling self-similar traffic, for example by using Point processes, modulated Poisson etc; Ryu gives a comparison of several methods [5-4]. An evaluation of the various methods was carried out, with the objective of finding one that was easily implemented in OPNET. In fact Ryu has written a tutorial on implementing a self-similar traffic model in OPNET using the raw packet generator (RPG) library module [5-5], but this was not implemented here since all that is needed is a simple packet source, whereas the RPG has a network, link and physical-layer stack. The method finally chosen was one based on a paper by Ulanovs [5-6], which sums the output from several ON-OFF sources.

The ON periods for each source are independent and identically distributed, as are the OFF periods, with the ON and OFF periods independent of one another. In order for the superposition of the ON-OFF sources to produce a self-similar source, the ON and OFF periods are determined from heavy-tailed distributions and the most widely used distribution for this purpose is the Pareto.

The Pareto distribution for a variable x is described by two parameters α and γ and is given by [5-7] as

$$P(x) = \frac{\alpha \gamma^\alpha}{x^{(\alpha+1)}} \quad (x \geq \gamma)$$

where α is the shape index and γ is the position index or minimum value of x .

The relationship between α and the Hurst parameter H is

$$H = (3-\alpha)/2$$

The design value for H is set at 0.75, in line with that found over a 4-year period of measurements on an internal network by Leland at Bellcore [5-2]. This sets $\alpha = 1.5$.

Values of γ have been chosen at 10 and 100 for the ON and OFF period distributions respectively, however these are somewhat arbitrary since the mean rate of packet generation depends both upon γ and upon the rate at which each source produces packets while ON. The shape of $P(x)$ against x is shown in figure 5-4 for $\alpha = 1.5$ and the two values of $\gamma = 10$ and $\gamma = 100$.

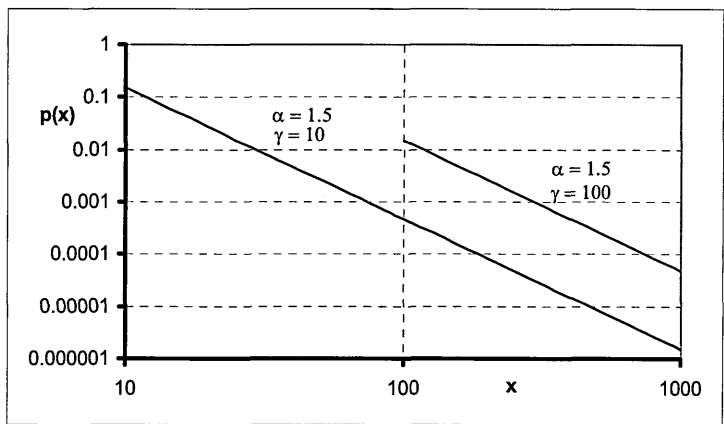


Figure 5-4 The Pareto distribution

This shows that the Pareto distribution is a straight line on log-log axes.

The mean of $P(x)$ is

$$E[x] = \frac{\alpha\gamma}{\alpha - 1}$$

$E[x] = 30$ and 300 for $\gamma = 10$ and 100 respectively, with $\alpha = 1.5$.

The ON-OFF sources each produce a packet stream when ON and produce nothing when OFF. In the model built here, when ON, the packets from every source have a constant length of 576 bytes and are generated at a constant rate λ . Figure 5-5 shows N packet sources followed by an ON-OFF switch associated with each source.

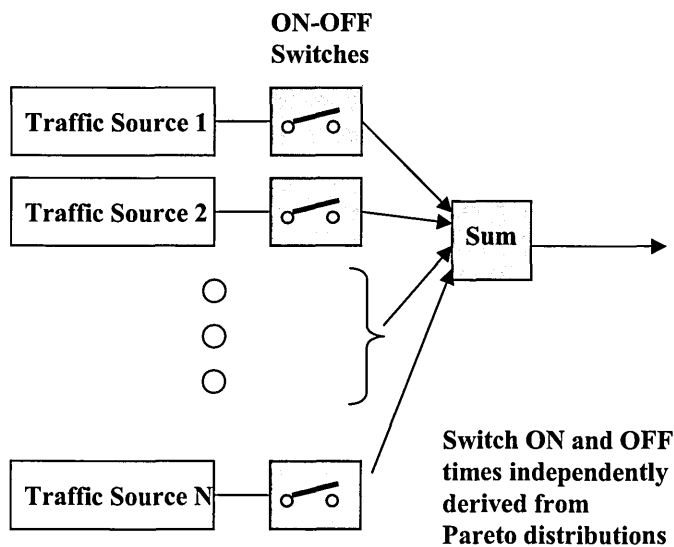


Figure 5-5 Arrangement of ON-OFF traffic sources

A constant generation rate is adopted from each source because it most closely resembles a user pattern, i.e. one user per stream.

Figure 5-6 shows the relationship between λ , ON and OFF times from each source.

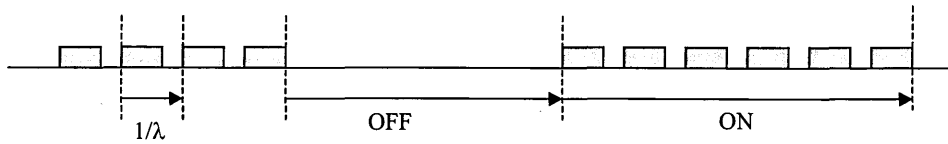


Figure 5-6 Packets are produced at a constant rate during ON times from each source

The pattern of packets generated during ON periods, combined with the OFF periods, determines the mean and variance of packet flow, i.e. the ‘burstiness’ from each source and hence from the whole generator. Distinct from this, H determines the rate at which the statistical properties degenerate with increasing aggregation over time. So H does not tell anything about the ‘burstiness’, but depends upon the distributions used for the ON and OFF periods and to some extent upon the number of sources. Therefore, in order to completely specify a self-similar source, it is necessary to specify (1) the pattern of packets generated by each source while ON, (2) the number of sources and (3) the value for H or α .

The paper by Ulanovs indicates that the value of self-similarity is not very sensitive to the number of sources, he shows that varying the number between 10 and 70 only gives about 10% change in H . The average of $N = 40$ sources is adopted here. This means that the mean generation rate from 40 superimposed sources is:

$$\lambda_{\text{tot}} = 40\lambda * \frac{ON}{ON + OFF}$$

where λ is the arrival rate from each source while ON.

The values of γ in the Pareto distributions can be fixed arbitrarily and the arrival rate from each source adjusted to get the desired λ_{tot} . So, assuming the adoption of mean ON and OFF times of 30 and 300 seconds, $\lambda_{\text{tot}} = 3.636\lambda$.

The model was constructed in OPNET by adding the outputs of 40 packet sources that are switched ON and OFF for periods drawn from the Pareto distribution. Full details of the code are included in Appendix A.

A continuous-time stochastic process $X(t)$ is said to be self-similar if, for any real, positive ‘ a ’, the processes $X(t_n)$ and $a^{-H}X(at_n)$ have the same statistical properties for all values of n , where H lies between 0.5 and 1. When $H = 0.5$ there is no self-similarity (such as for a Poisson process) and H approaches 1 as the degree of self-similarity increases.

Practically, statistical self-similarity means that the following equations hold (from Unalovs [5-6]):

$$\begin{aligned} \text{Mean } E[X(t)] &= E[X(at)] * a^{-H} \\ \text{Variance } \text{var}[X(t)] &= \text{var}[X(at)] * a^{-2H} \\ \text{Auto-correlation } R(t, \tau) &= R(at, a\tau) * a^{-2H} \end{aligned}$$

These equations can be interpreted to mean that the mean, variance and auto-correlation all fall off (or degenerate) over increasing time-scales at a rate dependent on the value of H . Of these three, the statistical property that is used to measure the degree of self-similarity here is the variance, where it is shown later that H can be estimated by measuring the variance degeneracy over different timescales, i.e. degrees of aggregation.

The equations above relate to a continuous process; in order to describe a method of aggregation for a discrete process, such as the case here with packets, consider a time series $X = X_n$ (n positive and real) and consider another time-series $X_n^{(m)}$ obtained by aggregating non-overlapping blocks of size m from the original series X . For instance, a block size $m = 10$ is formed by:

$$X_n^{(10)} = (X_{10n-9} + X_{10n-8} + X_{10n-7} + \dots + X_{10n-1} + X_{10n})/10$$

or, in general,

$$X_n^{(m)} = \frac{1}{m} \sum_{i=mn-(m-1)}^{nm} X_i$$

where n begins at 1 and is incremented in steps of 1 until the end of the data is reached.

Figure 5-7 illustrates the aggregation process.

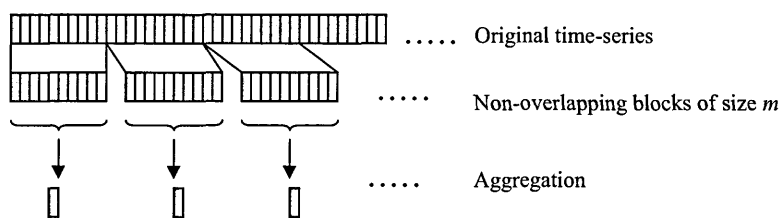


Figure 5-7 Illustration of aggregation process

Thus, the series $X_n^{(1)}$ contains the finest granularity and detail. As aggregation takes place, the detail reduces and, if the statistical properties remain constant despite the loss in detail, the process is said to be of a ‘fractal’ or ‘self-similar’ nature.

Process X is said to be self-similar with parameter β ($0 < \beta < 1$), if the following condition for variance is fulfilled:

$$\text{Variance } \text{var}[X^{(m)}] = \text{var}[X] / m^\beta \quad (5-3)$$

where β is related to H by $\beta = 2(1-H)$.

So, if there is any degree of self-similarity, the variance $\text{var}[X^{(m)}]$ drops by a rate that is slower than $1/m$ as $m \rightarrow \infty$, depending on the value of β . This is in contrast to stochastic processes where the variance reduces at a rate equal to $1/m$ and approaches zero as $m \rightarrow \infty$ (as shown by the central limit theorem – and also exhibited by the Poisson distribution).

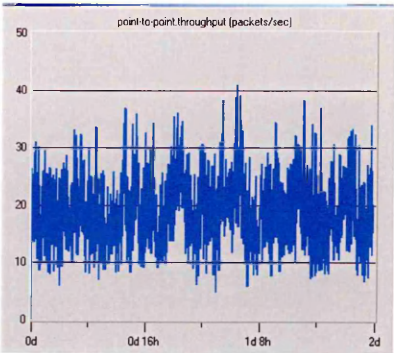
One way to measure H is to measure the rate at which the variance drops over increasing values of m and then measure the slope of β against m .

Proceeding by taking logs of equation (5-3),

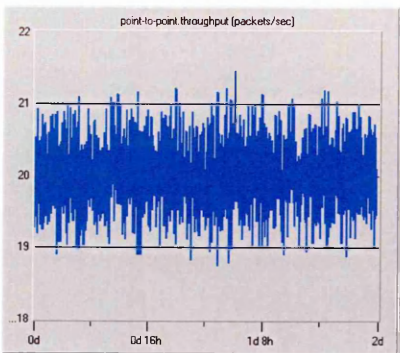
$$\log [\text{var}[X^{(m)}]] = \log[\text{var}[X]] - \beta \log [m]$$

Since $\log[\text{var}[X]]$ is a constant, a plot of $\text{Log} (\text{var}[X^{(m)}])$ against $\text{Log} (m)$ should be a straight line with a slope of $-\beta$. H can then be determined from $H = 1-\beta/2$.

The set of plots in figure 5-8 over the next couple of pages was taken from the output of the OPNET self-similar and Poisson traffic models, to enable a visual comparison of how the variance (or fluctuation) decreases with increasing aggregation. The aggregation over non-overlapping blocks was achieved in OPNET by making use of link probes in ‘bucket’ mode, where the bucket sizes can be set so that averages are taken over blocks of the raw data. The probes then averaged raw data points over non-overlapping blocks of m data points as required. Both traffic generators were set to a mean generation rate of 20 packets / second for this test. It can be seen that, as expected, the variance of the self-similar traffic drops more slowly than the Poisson traffic for increasing values of m .

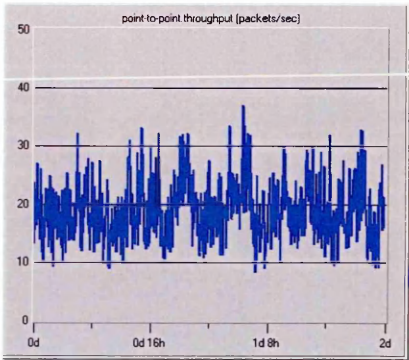


Self-similar

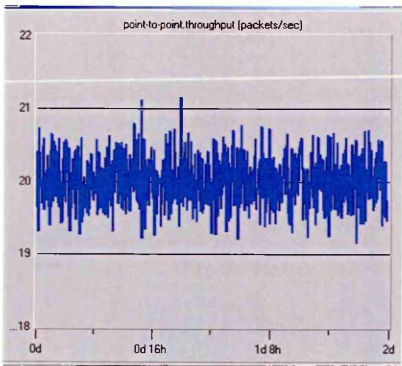


Poisson

Figure 5-8a Block size $m = 100$

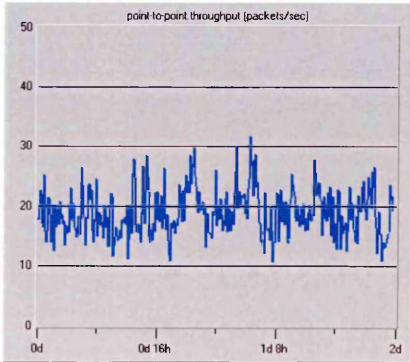


Self-similar

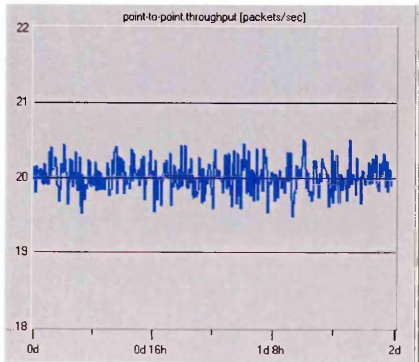


Poisson

Figure 5-8b. Block size $m = 200$

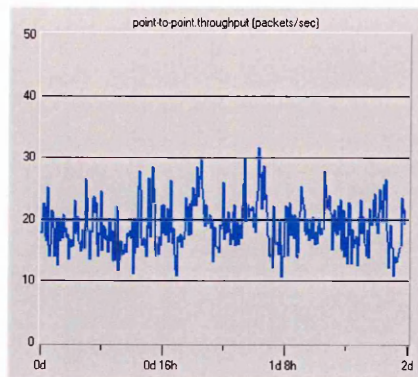


Self-similar

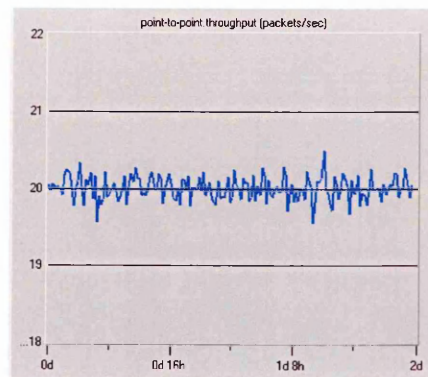


Poisson

Figure 5-8c. Block size $m = 500$

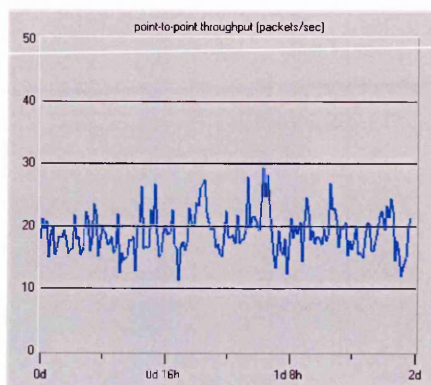


Self-similar

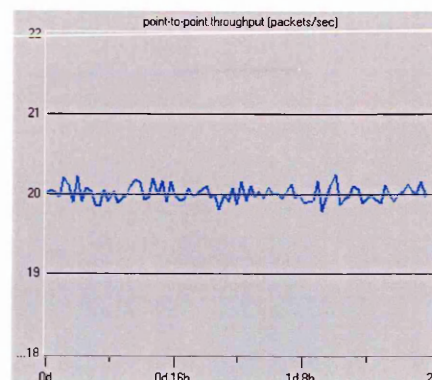


Poisson

Figure 5-8d. Block size $m = 1000$



Self-similar



Poisson

Figure 5-8e. Block size $m = 2000$

When performing these tests with different m , it is important to use the same raw data for each, hence the plots in figure 5-8 all have the same time-scale. Originally this was not done, the lower values of m were plotted using a shorter data set in order to save time, however with self-similar traffic there can occur large bursts of traffic at any time, which upsets the relationship when m is increased. Each of the plots in figure 5-8 was constructed from simulation runs of 200,000 seconds equating to around 2 days, in order to obtain at least 100 blocks at the highest aggregation level of $m = 2000$.

Looking down the plots it can be observed that the variance is inversely proportional to m ; but the rate of drop of variance is less with the self-similar source than with the Poisson source.

Figure 5-9 shows the a plot of the log of the variance $\text{Log}(\text{var}(X))$ for both self-similar an Poisson traffic, plotted against $\text{Log}(m)$ for m from 5 to 2000. Also included on this plot are reference lines of slope = -0.5 and -1.

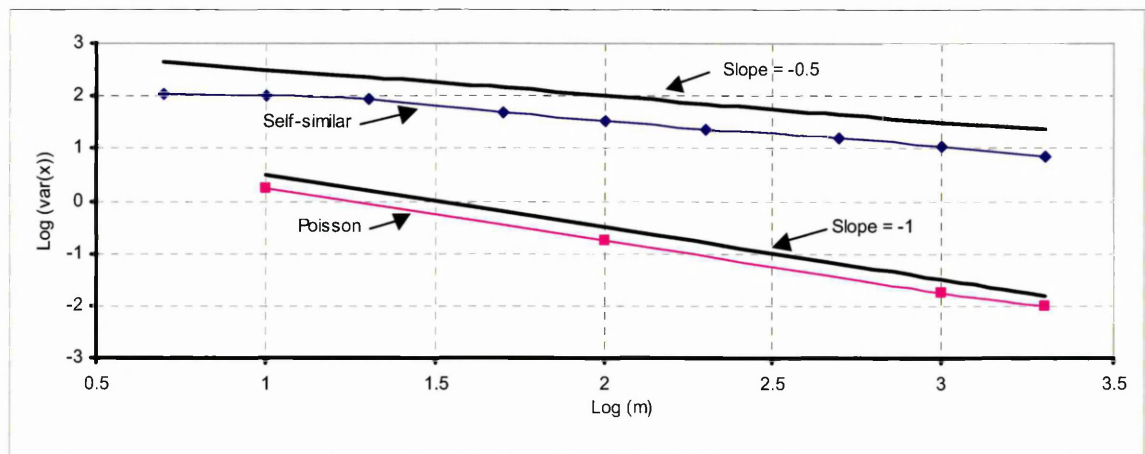


Figure 5-9 Plot of Log Var(x) against Log m.

Figure 5-9 shows that, for the self-similar source, the slope is very close to -0.5 over the range $\text{Log}(m) = 1$ to $\text{Log}(m) = 3$, except for the lowest value of m . According to the literature, the lowest values of m can be ignored when measuring this slope [5-6]. As the slope is $-\beta$, then $\beta = 0.5$ and $H = (1 - \beta/2) = 0.75$, which is the design value.

In the Poisson case, the slope = -1 so that $\beta = 1$ and therefore $H = 0.5$, corresponding to zero self-similarity as would be expected with the memoryless Poisson traffic.

Therefore the OPNET model has been calibrated for the correct amount of self-similarity, at least for $\lambda_{\text{tot}} = 20$ packets/s; there is no reason to suspect that it would not function correctly at other rates, so it is adopted with an adequate degree of confidence.

5.4 Queueing model

5.4.1 Description

The model adopted here is a simple first-in first-out (FIFO) queue and single server, together called a system, as shown in figure 5-10.

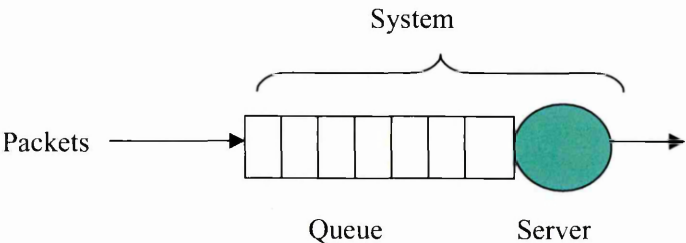


Figure 5-10 Simple queue and server system

This is a simple abstraction from a real network, which will have several queues and servers in a series / parallel path where congestion can occur. The abstraction is justified on the basis that in general a satellite link will be carrying traffic from several sources multiplexed together and it will be the dominant constriction in the path, since the satellite bandwidth will be at a premium. For this reason and for reasons of simplicity, just one queue is modelled here and it is left to further work to decide how well the findings apply to networks of queues. Jackson's theorem is used in some papers, for example Hwang et al [5-8] to represent a network of queues by a single queue, however the theorem cannot be invoked in this case because his equations only hold when the service rates have exponential distributions, which is not the case here. Constant length packets are used here, so that the service rate is deterministic with single server and the queue can be described as G/D/1; Kendall notation [5-9] is used throughout.

The queue in figure 5-10 was constructed in OPNET and calibrated using a slightly round-about two-step process, as follows:

- The queue was made M/D/1 in order to calibrate the waiting time moments (mean, variance)
- The queue was made M/M/1 in order to calibrate the delay pdf.

The Poisson packet generator described in section 5.3.1 was used to generate the Markov arrival process for both steps. The reason that a Poisson source was used rather than self-similar is that the queuing theory is far simpler with a Markov arrival process than it is with a general (G) process that would be necessary with a self-similar source. A constant packet length was used to force the service distribution to Deterministic for step 1, whereas an exponentially distributed packet length was used to force the service distribution to Markov for step 2.

Figure 5-11 illustrates the process adopted.

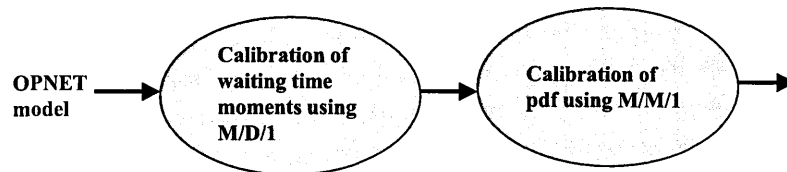


Figure 5-11 Calibration process using M/D/1 and M/M/1

The M/M/1 is not a good model to use with packets, since the exponential nature of the service time implies that some 'customers' take zero time to service and hence that some packets have zero length. The reason for using M/M/1 here is that its delay pdf is much easier to calculate from theory than the delay pdf for M/D/1; therefore the simulation model is first proven to produce a M/M/1 delay pdf that agrees with theory, then it is used to give a view of the delay pdf for M/D/1.

Although the M/D/1 is a much better model for packet network queues, its usefulness is limited because the arrival process with practical networks has a self-similar nature, rather

than Markov, as has already been pointed out. Nevertheless, the Markov arrival process is retained for this calibration step and for the first phase of the measurements, since the theory is relatively well understood and it serves as a bound of ‘no self-similarity’.

Having stated that the packet flows on real networks have a self-similar nature, the theory of queuing for self-similar flows is difficult, especially for the case where packet lengths are allowed to vary arbitrarily. In this case, the system in this study becomes G/G/1. Abate et al [5-10] are prominent in the field of queue theory and they describe some numerical solutions to G/G/1 equations originally developed by Kingman and Pollaczek and summarised by Kleinrock [5-11], which could possibly be developed for the self-similar traffic profiles used in this study given enough time. However it was considered not worth investing the time to develop the necessary theory, since the theory would produce exact probability density functions (pdfs) of the delay only if the exact stochastic characteristics for the traffic were available, which for a practical network they would not be.

The general form of the delay pdfs for Self-similar queue behaviour is obtained here through simulation, estimated from histograms; but with practical networks the accuracy of such estimations depends on the time available for the measurements and will vary from path to path and flow to flow.

So the starting point is the M/D/1, where queuing theory has been well documented. The waiting time moments can be calculated using the Takács recurrence relation [5-12]. This formula gives the queue delay only, the server delay has to be added to the queue delay to obtain the system delay. The recurrence relation, with slightly altered notation is:

$$\overline{W_q^k} = \frac{\lambda}{1-\rho} \sum_{i=1}^k \left(\frac{k}{i} \right) \frac{X^{i+1}}{(i+1)} W^{k-i}$$

$$\text{where } \frac{k}{i} = \frac{k!}{i!(k-i)!}$$

and where W_q = waiting time in the queue, k = moment, λ = mean packet arrival rate, ρ = load on queue ($= \lambda * E\{X\}$) and X = service time.

For $k = 1$ (first moment)

$$\overline{W_q} = \frac{\lambda X^2}{2(1-\rho)} \quad (5-4)$$

This is the mean waiting time in the queue and is also known as the Pollackzek-Kinchin (PK) mean waiting time formula. In fact, although the PK equation is derived here for M/D/1, it also holds for queues with general service rate distributions (M/G/1). Here it is derived from a recurrence relation; its derivation from first principles is given in Appendix B for interest.

Proceeding to calculate the variance:

For $k = 2$ (second moment)

$$\begin{aligned}\overline{W_q^2} &= \frac{\lambda}{1-\rho} \left[\left[\frac{2X^2}{2} \right] W_q + \left[\frac{X^3}{3} \right] W_q^0 \right] \\ &= 2\overline{W_q^2} + \frac{\lambda X^3}{3(1-\rho)}\end{aligned}$$

The variance of the waiting time in the queue is calculated from the second moment by subtracting the square of the mean:

$$\sigma_q^2 = [\overline{W_q^2} - \overline{W_q}^2] = \overline{W_q^2} + \frac{\lambda X^3}{3(1-\rho)} \quad (5-5)$$

Note that equations (5-4) and (5-5) above give the mean and variance of the delay pdf but not its shape. The shape of the pdf is important because it relates the mean and variance to the percentiles and quantiles for delay which, from chapter 4, is the way delay variation is quoted as a QoS parameter by the standards bodies. For this reason the theory for delay pdf shape is now explored a little further.

It has already been stated that the shape of the delay pdf is much easier to calculate with the M/M/1 queue than with M/D/1 and this statement can now be justified as follows. Consider the Laplace transform of the delay pdf for a system (queue plus server) with a general service time distribution (M/G/1) which is, from Kleinrock [5-13] (with modified notation),

$$a(s) = B(s) * \frac{s(1-\rho)}{s - \lambda + \lambda B(s)} \quad (5-6)$$

Where $a(s)$ is the Laplace transform of the wanted delay pdf and $B(s)$ is the Laplace transform of the service time pdf.

With the M/D/1 system, the service time is single-valued (i.e. deterministic), which means that its pdf is a unit impulse at $t = 1/\mu$. This is a delayed impulse function, which has the Laplace transform $B(s) = e^{-s/\mu}$.

Substituting this into (5-6) gives

$$a(s) = \frac{s(1-\rho)e^{-s/\mu}}{s - \lambda + \lambda e^{-s/\mu}} \quad (5-7)$$

The inversion of this transform proves to be very difficult. An analytical solution could not be found so numerical inversion was considered. However this generally involves a series that needs to converge and the accuracy of the inversion depends upon the number of terms in the series. In order to converge at all, it is a necessary condition that the transform is well-behaved

(i.e. no sudden changes); see for example Abate et al on the Laguerre method for Laplace Transform inversion [5-14]. As the transform in (5-7) contains an impulse, it is not well behaved and so it is not possible to guarantee convergence.

With the M/M/1 system however, the pdf shape is much easier to determine. The service rate is exponentially distributed ($= \mu e^{-\mu t}$) which, by inspection, has the transform $B(s) = \mu / (s + \mu)$ (where μ = mean service rate).

Making this substitution for $B(s)$ into (5-7), manipulating and putting $\lambda = \mu\rho$ gives

$$a(s) = \frac{\mu(1-\rho)}{s + \mu(1-\rho)}$$

The inversion of this Laplace transform can be performed by inspection and results in the pdf shape

$$a(t) = \mu(1-\rho)e^{-\mu(1-\rho)t} \quad (5-8)$$

This equation is used in the next section to calibrate the OPNET system model. The OPNET model is then also used to give a shape of $a(t)$ for M/D/1, for which an equation could not be found.

5.4.2 Calibration of the OPNET model

The calibration results from the OPNET queue model will now be compared with the relevant equations for delay moments and pdfs.

In order to calibrate the OPNET queuing model, a simple model illustrated in figure 5-12 was constructed. It consists of a User_traffic_server, a queue and a User_traffic_client.

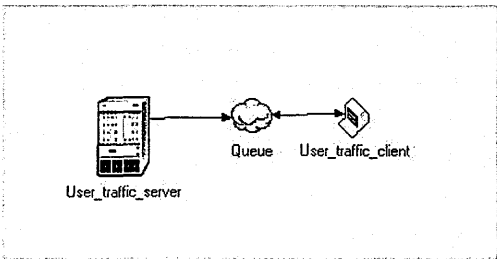


Figure 5-12 Simple OPNET queue and server system

The User_traffic_server generates packets in selectable format (packet size and inter-arrival time) and applies them to the queue. At the output of the queue, a link carries the packets away from the queue at a defined bit-rate. The User_traffic_client accepts the packets and is fitted with a probe to measure the throughput and the end-to-end delay of the packets.

Using the model in figure 5-12, the service rate μ is set at 27.778 packets / second, because this results in a bit-rate from the server of 128kbit/s with 576 byte packets. The mean packet arrival rate λ is varied to give a range of ρ , bearing in mind that $\rho < 1$ for stability.

Figure 5-13 shows an example plot of system delay (i.e. queue plus server) against time for $\rho = 0.95$ and model = M/D/1.

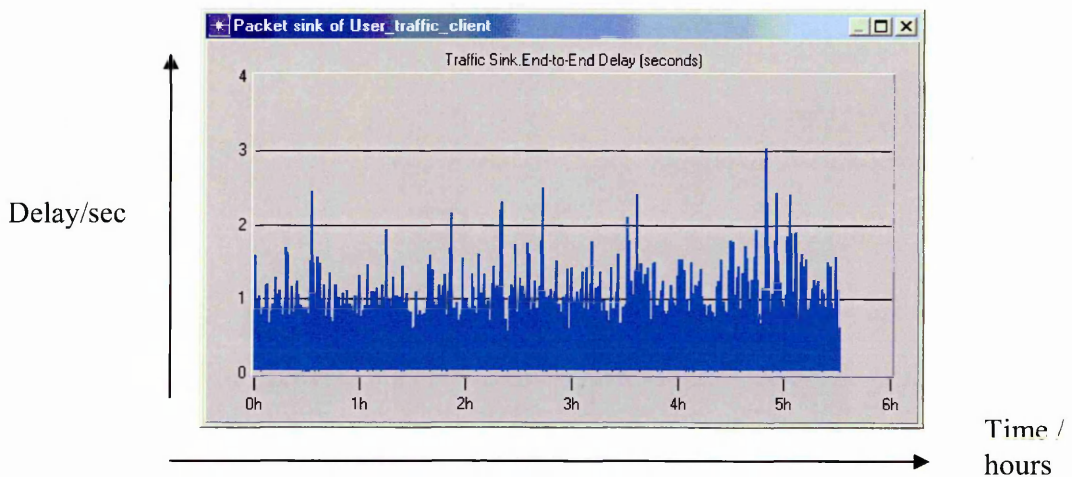


Figure 5-13 Example time-series delay for $\rho = 0.95$, M/D/1 system

The mean system delay in figure 5-13 is 0.34 s and the variance is 0.119 s^2 . The minimum delay is 0.036 s which is the time taken to service the 4608-bit packets at 128 kbit/s. This occurs when an arriving packet finds the queue empty and enters straight into service. The conversion to queue delay can be easily achieved by subtracting 0.036s from each system delay value.

In order to compare the OPNET model in figure 5-12 with theory, the mean and variance of the simulation runs are compared with the mean and variance from theory, equations (5-4) and (5-5) respectively, for the various values of ρ . A constant delay of 0.036s has been added to the values from equation (5-4) to give theoretical system delay. Figure 5-14 shows the results of the simulations as 'crosses' on the theoretical plots. It can be seen that the simulation results and theoretical results are in very good agreement.

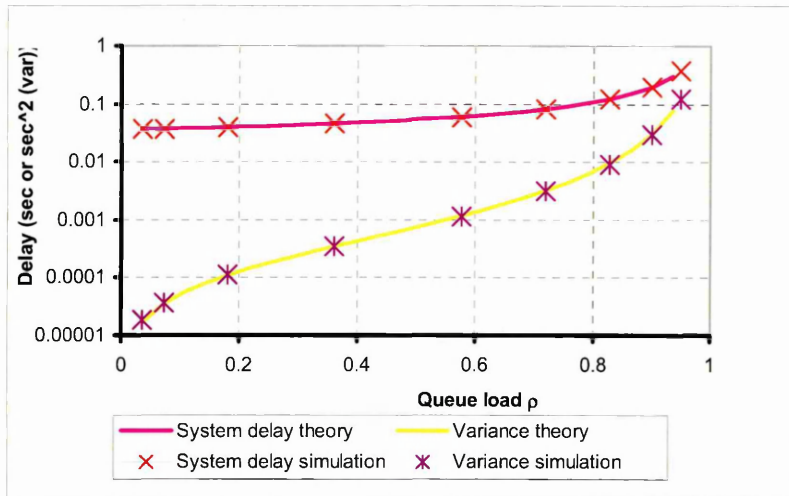


Figure 5-14 Comparison of simulation results with theory – delay mean and variance for M/D/1 system

An observation while running these simulations was that the simulation time is required to be longer for higher values of ρ for the simulation results to converge, which was considered important for this calibration process. In order to produce figure 5-14, a simulation time of 10,000 seconds was used for $\rho \leq 0.5$ and 20,000 seconds for $\rho > 0.5$. The simulation time was considered long enough if, when doubled, the results change less than 1%.

These rather impractical measurement times are brought about because queue theory is developed from differential equations when all transients have died down (see Appendix B) and it can take many hours to achieve this steady state in practice when ρ is close to 1. The implications of long measurement times, and some discussion on shortening them, are presented in chapter 6.

The initial conclusion from figure 5-14 is that the OPNET queue and server system is behaving properly. The next step is to look at the waiting time histograms and pdfs and compare the M/M/1 pdf shape with equation (5-8).

5.4.3 Waiting time distribution

Figure 5-15 shows the normalised histograms from simulation runs of M/M/1 for $\rho = 0.3$ and for $\rho = 0.7$ (service rate mean $\mu = 27.78$). These were constructed from system delay measurements taken by the OPNET model in figure 5-12 using about 40,000 data points. Also on the plot are theoretical pdf curves from equation (5-8).

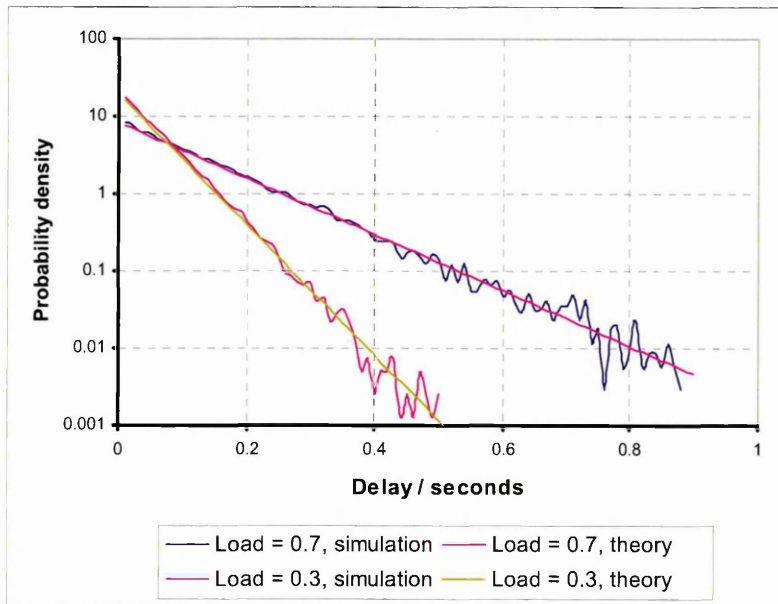


Figure 5-15 Comparison of simulation results with theory, system delay pdf for M/M/1

Figure 5-15 shows that the M/M/1 simulation results are a good match to the theoretical formula (5-8). The simulation plots get noisier for higher values of delay where there are fewer data points in the histogram bins.

Now that the model has been calibrated, it is interesting to see what the pdfs of system delay for M/D/1 look like for various queue loads, remembering that these could not be obtained analytically by inversion of equation (5-7). Figure 5-16 shows plots of the pdfs derived from histograms for $\rho = 0.3$ and 0.7 (still with $\mu = 27.78$) along with fitted curves.

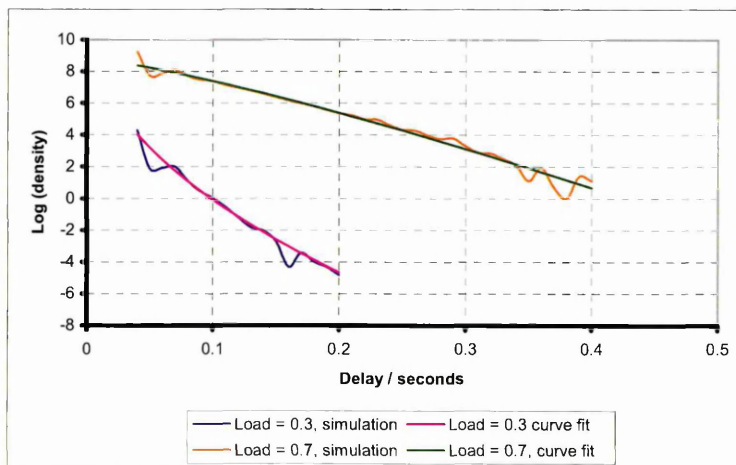


Figure 5-16 M/D/1 system delay pdfs from simulation histograms, with fitted curves

In order to fit the curves in figure 5-16, the log of the density is taken first so that the least-squares procedure does not give too much weight to the tail. Curves are then fitted such that

$$\text{Log}(\text{density}) = \text{Log}(a) - b d^c$$

Where a , b and c are curve parameters and d is the delay. Least-squares fit values for $\text{Log}(a)$, b and c are given in table 5-1 for values of $\rho = 0.3$ and 0.7 .

Queue load ρ	$\text{Log}(a)$	b	c
0.3	11.5	35	0.48
0.7	8.85	25.5	1.24

Table 5-1 Least squares fit for pdfs of M/D/1

As $c \neq 1$, these curves are not simple exponentials, which is why the inversion of the Laplace transform is difficult for the M/D/1 waiting time distribution. Iverson and Staalhagen [5-15] give a number of equations for the M/D/1 waiting time distributions that confirm their non-simple nature; some of the equations contain fractional exponentials and others have infinite series solutions. As promised earlier, the curves in figure 5-16 give a view of the inversion of equation (5-7) for $\rho = 0.3$ and 0.7 .

In conclusion, the OPNET queue that has been created agrees with the theory for both the moments for M/D/1 system delay and for the pdf for M/M/1 system delay. It is therefore assumed that we can proceed with confidence with this model. The model has also been used to give a view of the M/D/1 system delay pdf.

Now that OPNET models of Poisson and self-similar packet generators and of a queue have been produced and calibrated, they can be used to perform measurements of QoS parameter estimation accuracy using trace packets. Figure 5-17 shows the development of the OPNET model to include the facility to introduce trace packets to accompany user packets.

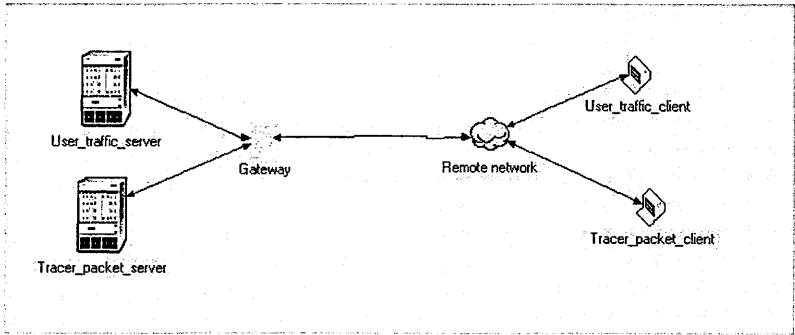


Figure 5-17 OPNET model for User plus Trace packets

The `User_traffic_server` and `User_traffic_client` in figure 5-17 are the traffic source and sink as in figure 5-12. The additional items in figure 5-17 are a server and client for the trace packets. The simple queue has been replaced by a Gateway and Remote network, with the ‘satellite’ link in-between that passes them at a constant rate of 128kbit/s. The Gateway places the user and trace packets into a FIFO queue¹ and then transmits packets from the head of the queue. The Remote network is basically a switch that routes packets to the correct client; the

¹ In this additional code, sub-queues are also created with one sub-queue having service priority over the others in order that event traffic (such as alarms) are given top priority. Although tested, this was not used.

presence of this switch adds a further small fixed delay to the path but otherwise does not interfere with the packets. Within each client are probes that collect the end-to-end packet delay times and cumulative packet counts, so that delay and loss statistics can be calculated.

5.5 The test network

5.5.1 Description

The test network consists of a satellite link emulator, with a dominant queue ahead of it, to resemble the single queue and server system in figure 5-17, along with a test packet generator capable of generating trace packets concurrently with Poisson or self-similar user traffic, two routers and other test equipment. A block diagram of the test network is shown in figure 5-18.

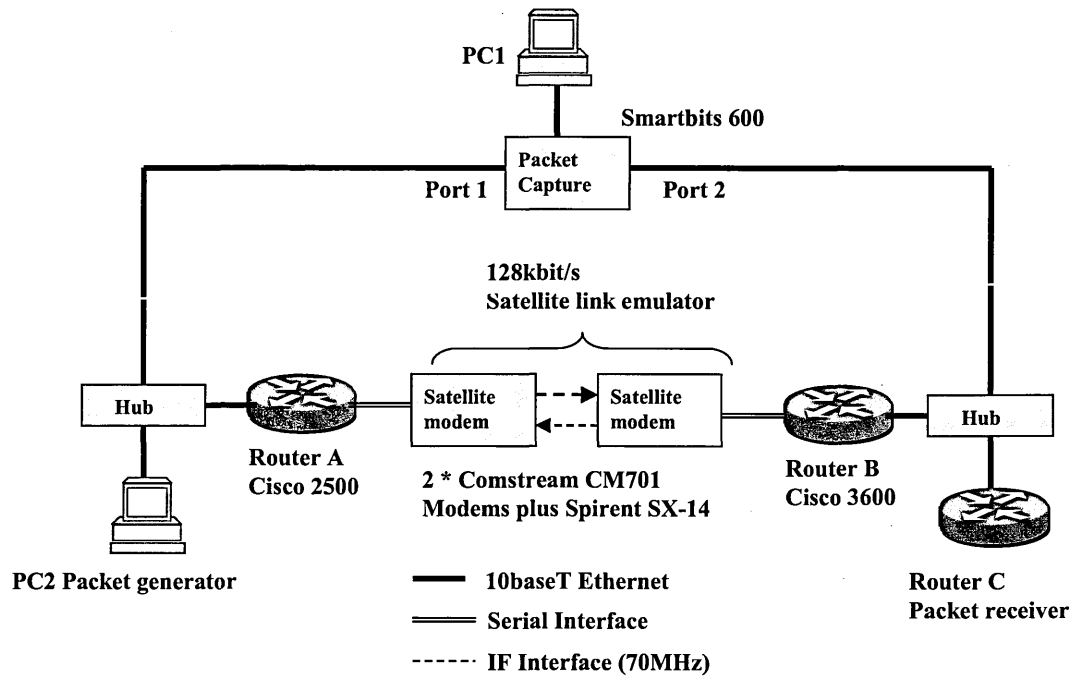


Figure 5-18 Test network diagram

Describing the test network from the top of figure 5-18, the ‘packet capture’ is an item of test equipment (Smartbits 600) that has two ports and here was used to capture packets arriving at both ports together, recording their length, contents and inter-arrival times, i.e. the time between the start of one packet and the start of the next. It does not record absolute arrival times.

PC1 runs Windows 2000 and is used to configure the Smartbits 600 using the SmartWindow client. PC1 was also used to download the packet capture data from the Smartbits 600. PC2 is dedicated to running a packet generator application called RUDE that is capable of any traffic distribution and is described in more detail later. Router C acted as a packet sink; a router was used for this purpose rather than a PC, since a PC in this position tended to send replies onto

the network that complicated the packet capture. Figure 5-19 is a photograph of the test network equipment.

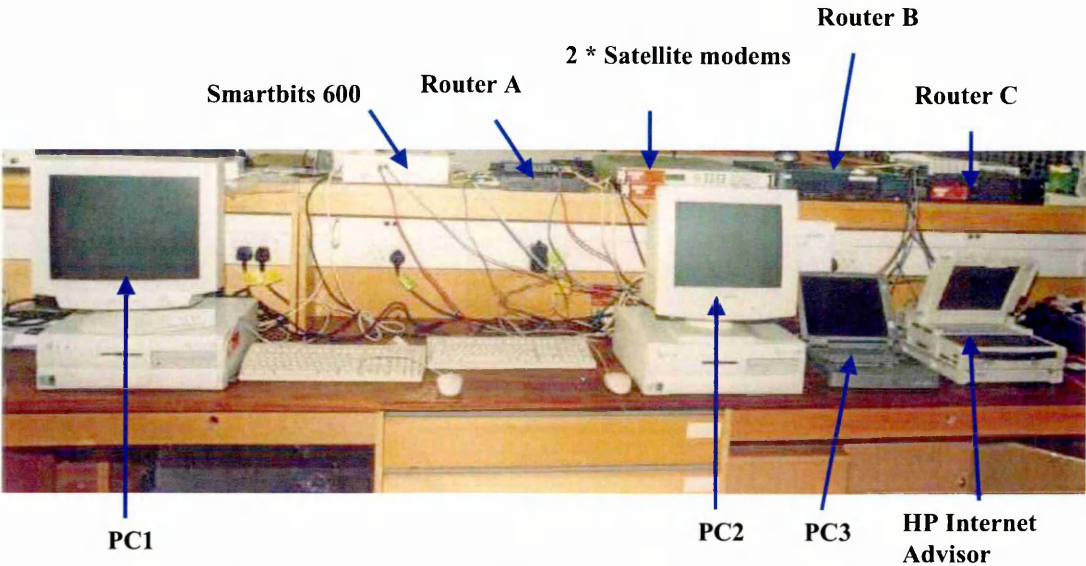


Figure 5-19 Photograph of test network equipment

There are two pieces of equipment in the photograph that do not appear in figure 5-18. These are PC3, which was used to configure the routers via their console ports and the Internet Advisor, which is a packet analyser that was used to confirm encapsulation overheads and header structures.

In all tests it proved necessary to filter the wanted packets from the logs taken with the Smartbits 600, since there were other packets present on the network such as ARP broadcasts. All filtering and processing was carried out post-capture using Microsoft Excel.

When the test network was first set up, the Smartbits 600 was used to generate the packets in addition to capturing them, however it was soon discovered that this instrument cannot generate packets to a distribution like Poisson or self-similar, it can produce only continuous packets or simple bursts. It was then decided to investigate the possibility of using a PC-based packet generation tool. After some private discussions and research on the Internet it became clear that there are freely downloadable tools to generate test packets, but these all ran on Linux. The Linux operating system is more conducive to running packet generation applications than Windows, because of the more flexible way it deals with sockets. Therefore some knowledge of Linux had to be gained and, as a result, PC2 runs Red-Hat Linux and an application called RUDE that is used to generate test packet streams.

RUDE, which is free software downloaded from Sourceforge [5-16], was set to generate packet flows using its TRACE option (not to be confused with the concept of trace packets). Using this option, RUDE reads down a text file that contains two columns, the first contains the required packet length and the second contains the time interval between the current packet

and the previous packet. Using two such files, RUDE was used to generate two flows together as follows:

- a user packet flow with such Poisson or self-similar characteristics and variable packet length
- a trace packet flow with a constant inter-arrival time and packet length of 64 bytes

The trace packets use a constant distribution, so that the TRACE file for these need only consist of one entry. There is an option within RUDE to generate constant rate packets using the CONSTANT option in the script, but this was not used since it only works for inter-arrival times of 1 second or less.

The user packet TRACE files were constructed by running the OPNET simulation model with the packet generator set to give the required distribution and rates. This was achieved by modifying the Gateway node process model so that it calculated and collected packet inter-arrival times as a statistic and then a probe was created to record the statistic. The OPNET model was then run for up to 10,000 seconds for the required range of user packets and the inter-arrival times saved to text files. These were edited to remove OPNET text, to make sure all of the numbers are in real format (RUDE trips up on numbers in exponential format) and to insert packet lengths. For use by the TRACE option in RUDE, the text files were then copied to the same directory as RUDE on PC2. An example of a TRACE file is given in Appendix C.

The packet lengths specified in the TRACE text files do not include the IP header, therefore if 576 byte IP packet lengths are needed (as here), then 556 bytes have to be specified; RUDE then generates 576 byte packets (including 20 byte headers) and these are transmitted on Ethernet from PC2 with a total frame length of 602 bytes. Figure 5-20 shows the construction of an Ethernet frame with its 26 overhead bytes.

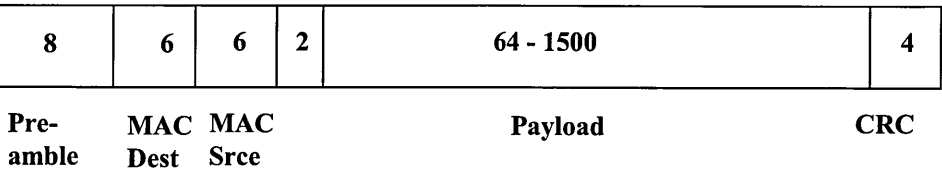


Figure 5-20 Construction of IEEE802.3 Ethernet frame

In figure 5-20, the pre-amble is present to allow the receiver to synchronise a clock to the packets as they arrive (Ethernet is asynchronous). The MAC Destination and Source fields are 6 bytes each and the 2-byte field gives the frame type, which is set to 0800 for IP. The payload is between 64 and 1500 bytes, which is where the IP packet is inserted. If an IP packet is longer than 1500 bytes it must be fragmented in order to fit into standard Ethernet frames. The 4-byte CRC detects errors anywhere in the frame. The total overhead added by the Ethernet frame is therefore 26 bytes, which when added to 576 bytes gives the length of 602 bytes observed.

Router A in figure 5-18 is connected via a serial interface to the satellite modems, with the router configured as DTE (clock produced by the modem). It is in this router that a queue will develop where packets await transmission over the satellite link since, at 128kbit/s, this is the dominant constriction in the end-to-end path. Packets that arrive to find the queue full are discarded.

Router B accepts the packets from the satellite modems and routes them to the subnet containing Router C, which is the recipient of the packets. The router configurations used in figure 5-18 are given in Appendix C.

The satellite modems emulate the satellite link running at 128kbit/s (configured by the modems) which is the same as that used in the simulations in chapter 5. They were set to have a nominal FEC and modulation type: 1/2-rate k=7 convolutional code (no outer code) with QPSK. Most of the tests were performed with a satellite emulator for cost reasons but some basic checks were performed with a live link for which additional components are added between the modems.

Figure 5-21 to figure 5-24 show these additional components.

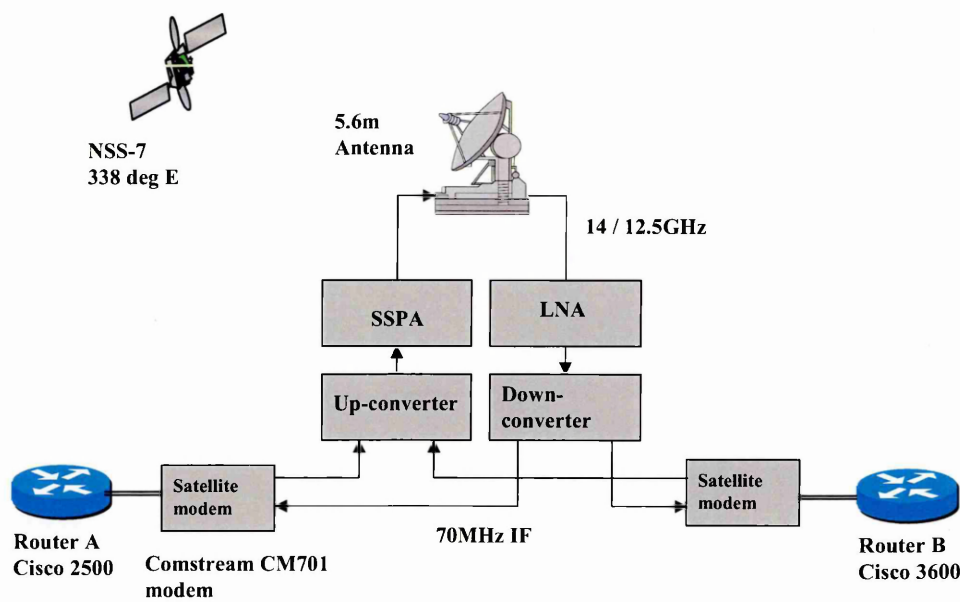
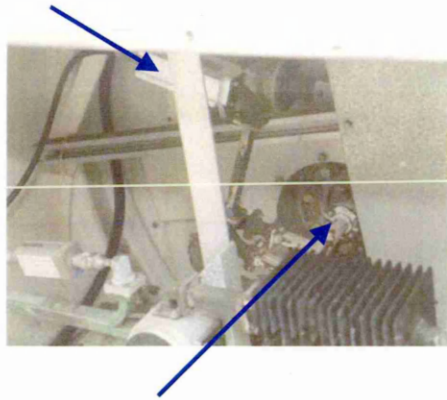


Figure 5-21 Additional components for the live satellite checks



Figure 5-22 Photograph of 5.6m antenna

LNA



Rear of antenna feed

Figure 5-23 Photograph of rear of antenna showing Low Noise Amplifier (LNA)

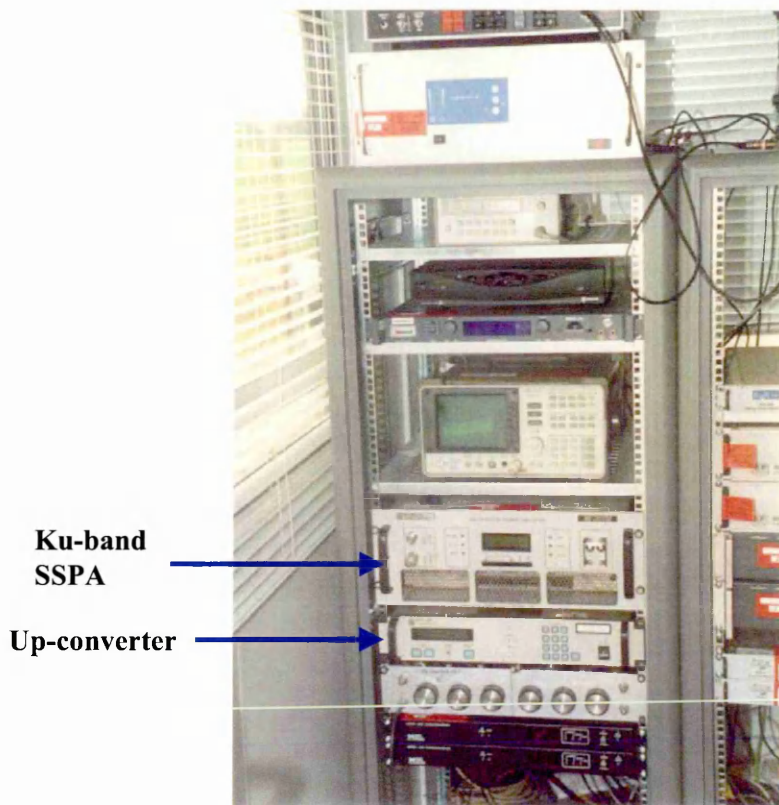


Figure 5-24 Equipment rack inside earth station

The satellite used was New Skies Satellites (NSS) flight 7 [5-17], with a Ku-band transponder on loop-back to the Europe / Middle East Spot. It was necessary to transmit two loop-back carriers to achieve a duplex connection. The two 128kbit/s carriers were set-up in accordance with NewSkies procedures and left to run for a few hours on soak test before any measurements were made.

5.5.2 Network set-up and test

Routing and ping tests

There were two anomalies noted when setting up the network routing:

- the routers automatically built ARP tables for the PCs, but for some reason the Smartbits 600 did not discover the MAC addresses of the router ports to which it was connected, therefore it was necessary to manually enter these into the Smartbits 600 port set-up,
- PC2 was set up manually using the commands 'ipconfig' and 'route' in order to set the Eth0 IP address and to set up a static default gateway route so that packets destined for foreign subnets generated by the application are sent towards router A. It appears that Linux does not build this automatically from knowledge of the gateway IP address.

When the routing was sorted out, 'Ping' tests end-to-end from PC2 to Router C gave a round-trip time (RTT) of 81–82 ms without the live satellite link present. Similarly, sending Pings from Router A to Router B also gave an RTT of 81-82ms, therefore this delay is mostly due to the modems which give around 41ms delay in each direction. The RTT with a live satellite link was measured at 613ms; this can be factored as $2 * 41$ ms for the modems plus $2 * 266$ ms for the satellite hops.

Measurement of queue length

IP packets generated by PC2 will be queued at Router A, a Cisco 2650, if they arrive faster than the satellite modems can transmit them. Although the queue load $\rho < 1$, the packets are produced by Poisson or Self-similar processes that are bursty in nature and therefore a queue will be present for some of the time.

Routers have more than one queue, however the dominant one in router A will be the queue associated with the serial output that is connected to the satellite modems. It was thought worthwhile to measure the length of this queue because the literature from Cisco for this router is not clear on this topic. After running some tests with user and trace packets, which are different lengths, it was realised that this queue had to be set to FIFO with tail discard; by default it is weighted-fair, which may change the order of packets in different flows in order to more efficiently utilise the bandwidth. Since trace packets are being used here to estimate user packet QoS parameters, it is necessary to maintain their order.

This highlights one possible disadvantage of using trace packets as estimators: packet order must be maintained. In a complex network, packets may be sent on different routes, which would upset the estimations. It is preferable to insert trace packets at a node close to the satellite link so that different routes are not possible and packet order is preserved.

To measure queue size, a burst of $1000 * 602$ -byte packets was launched from PC2 to router A at a rate of 100 packets/s, so that the burst length is 10 seconds. Within these packets are 576-byte IP packets addressed to PC3. A total of 325 packets arrived at PC3. At the end of the 10 second input burst, the server has removed 278 packets from the queue ($\mu = 27.8$ packets/s), so at that instant the queue must have held $325 - 278 = 47$ packets which is $47 * 576 * 8 = 216.6$ kbits (assuming that the queue does not hold link-layer overhead bytes). This was rounded to 215 kbits for use in the simulations.

Unexpected delay jitter

During the queue-length test described above, it was expected that the 313 packets would be evenly spaced upon arrival at the remote LAN, since they have been transmitted away from the queue at router A at a constant rate of 27.78 packets/s. This corresponds to an inter-arrival time of 0.03599 seconds. In fact they had the expected mean arrival time, with a variance of about $4 * 10^{-10} \text{ s}^2$, but when a histogram of the delay was plotted, there was a surprising double-peak as shown in figure 5-25.

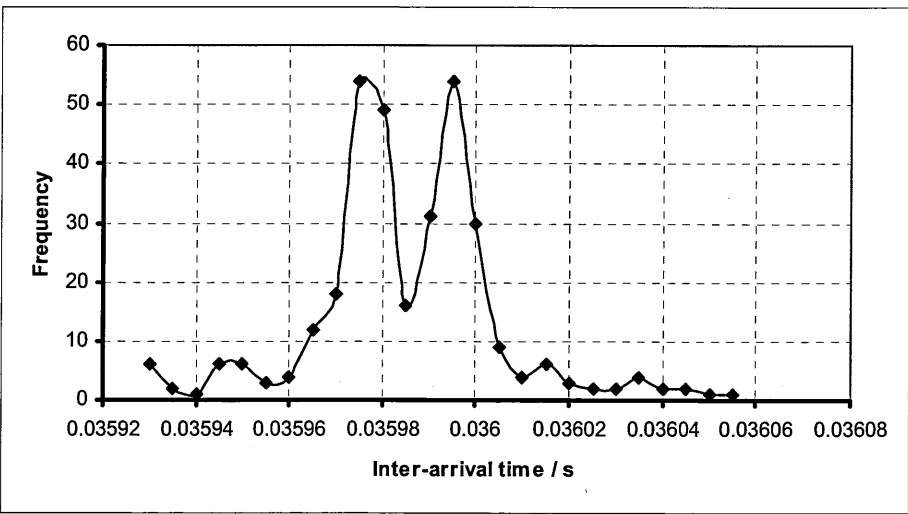


Figure 5-25 Histogram of inter-arrival times of test packet burst

It was considered worthwhile investigating the cause of this double peak, since it indicates that delay jitter is present and this could affect the accuracy of delay estimation. The distance between the peaks in figure 5-25 is about 1.8×10^{-5} s. It was thought that the jitter may be a function of packet length, so the packet length was changed from 576 bytes to 1500 bytes and the test re-run. This resulted in an increase in the distance between the peaks to about 0.019 s, with a variance of $7.53 \times 10^{-5} \text{ s}^2$. The variance therefore increased by a factor of 10^5 for a 3 times increase in packet length. This high sensitivity to packet length points to the routers being the cause, since the satellite modems have no knowledge of the packet boundaries. The sensitivity to packet length is also a concern because in this study, short trace packets are being used to estimate the delay to longer user packets.

It was guessed that the queuing process may have been responsible in some way, so a test was run where 500 * 1500 byte packets were sent from PC2 at a very low rate (8 packets/s) so that no queuing takes place. The inter-arrival time histogram from this test is shown in figure 5-26.

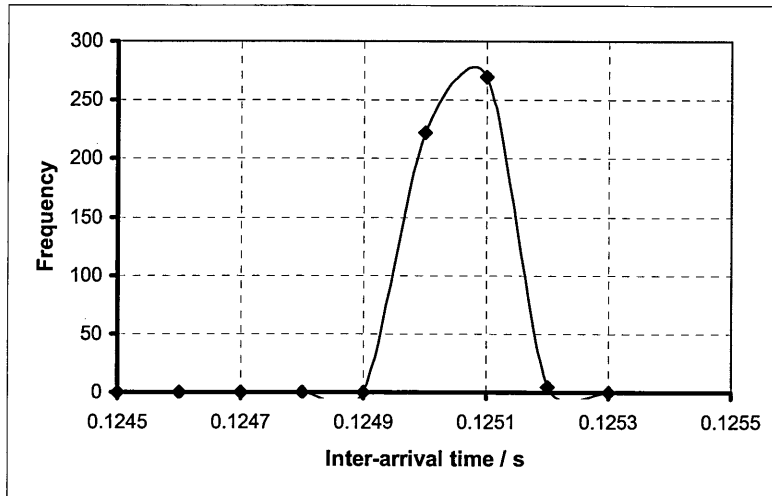


Figure 5-26 Histogram of inter-arrival times with no queuing

In figure 5-26 the mean inter-arrival time is 0.125s (as expected) and the variance is $7.9 \times 10^{-9} \text{ s}^2$. It shows an expected normal distribution with no jitter; absence of the jitter when no queueing takes place shows that something in the queuing or discard process is the cause, so a further test was performed by sending a short burst of packets very quickly so they had to queue, but not so that the queue filled. This resulted in re-appearance of the jitter, so it is concluded that the queuing process is the cause, rather than the discarding process.

It was then discovered that routers use distributed processing², with a processor dedicated to each interface and a central processing architecture to perform the core routing functions. The queue is formed at the interface where packets exit the router, so that the processor at this interface is responsible for both managing the queue and for sending the packets. It is postulated that, when this processor is occupied by moving packets up the queue, there is a slight delay in sending the packet at the head of the queue. It seems that more queue processing is required on longer packets and therefore the jitter is greater for these. This can be tested by looking at a sample of actual packet inter-arrival times. Below is an extract from the log-file of packet inter-arrival times (in seconds) from the set that was used with 1500 byte packets.

```
0.114626
0.095094
0.114601
0.095104
0.114618
0.095078
0.114662
0.095081
0.114562
```

² Through private discussions with router experts at BT

0.095106

This pattern repeats in similar fashion throughout the file, with minor variations. The jitter appears very much to be between alternate packets, which is evidence to support the theory that the interface processor is alternately processing the queue and sending packets. This pattern also explains why the histograms are highly symmetric.

The extent of the jitter is probably dependent upon the interface processing ability of the router and will therefore vary between routers. High-end routers are likely to exhibit lower jitter; the Cisco 2600 used in the test-bed is fairly low-cost and typical of what might be found in small remote networks. It was decided not to further investigate the non-linear relationship for two reasons:

- although the variance increases with packet length at an alarming rate, with 1500 byte packets (which are the largest likely to be encountered) it is still at least two orders of magnitude less than the variance due to the queue when the load is above about 0.3 (see chapter 6), so that the results of this study are not significantly affected,
- the relationship between jitter and packet length is evidently dependent on router processing, so would vary between routers and it is more suited to a study on router behaviour. It is logged for further work.

5.6 Summary of chapter 5

Poisson and Self-similar traffic models used in this study have been described, along with a queueing model and a test network. Each has been calibrated either against a mathematical model or against one another, so that sufficient confidence is considered to exist in the models for the next chapter, where results are presented of the use of trace packets to estimate user packet delay and loss. The presence of the live satellite link added an expected constant delay but made no discernible difference to the delay variation or packet loss since no fades were encountered during the tests.

5.7 References for chapter 5

- [5-1] OPNET Technologies website is <http://www.opnet.com/products/modeler/home.html> Link checked 7th August 2004
- [5-2] W Leland, W Willinger, M Taqqu, D Wilson. *On the self-similar nature of Ethernet Traffic*. ACM Sigcomm 93, San Francisco 1993.
- [5-3] L. Kleinrock. *Queueing Systems, Volume 1: theory*. ISBN 0-471-49110-1. John Wiley and Sons, 1975. Page 60.
- [5-4] B Ryu, S Lowen. *Point Process Models for Self-Similar Network Traffic, with Applications*. Stochastic Models, 1998.
- [5-5] B Ryu *A Tutorial on Fractal Traffic Generators in OPNET for Internet Simulations*. Hughes Research Laboratories, 2000.
- [5-6] P Ulanovs and E Petersons. *Modelling Methods of Self-similar Traffic for Network Performance Evaluations*. Scientific Proceedings of RTU, Telecommunications and Electronics 2002.
- [5-7] J Gordon. *Pareto Process as a Model of Self-Similar Packet Traffic*. IEEE Global Telecommunication Conference, Vol 3, 1995. Pages 2232 – 2235.
- [5-8] Y Ha Hwang, S Wook Chung, G-H Lee and Y Kim. *A Performance Analysis of TMN Systems Using Models of Networks of Queues, Jackson's Theorem and Simulation*. ETRI Journal, Volume 24, Number 5, October 2002. Pages 381 – 390.
- [5-9] S Bose. *An Introduction to Queueing Systems*. ISBN 0-306-46734-8. Kluwer Academic, 2002. Pages 20 – 22.
- [5-10] J Abate, G Choudhury and W Whitt. *Calculation of the GI/G/1 waiting-time distribution and its cumulants from Pollaczek's formulas*. Contribution to the 1993 special issue of AEU on Teletraffic Theory and Engineering, December 1st 1992.
- [5-11] L. Kleinrock. *Queueing Systems, Volume 1: theory*. ISBN 0-471-49110-1. John Wiley and Sons, 1975. Pages 299 - 303.
- [5-12] L. Kleinrock. *Queueing Systems, Volume 1: theory*. ISBN 0-471-49110-1. John Wiley and Sons, 1975. Page 201.
- [5-13] L. Kleinrock. *Queueing Systems, Volume 1: theory*. ISBN 0-471-49110-1. John Wiley and Sons, 1975. Pages 199 - 200.

- [5-14] J Abate, G Choudhury and W Whitt. *On the Laguerre Method for Numerically Inverting Laplace Transforms*. AT & T Bell Laboratories, April 1995.
- [5-15] VB Iversen and L Staalhagen. *Waiting time distribution in M/D/1 queueing systems*. Electronics Letters, 9th December 1999. Vol. 5 No. 25.
- [5-16] RUDE is available from the Internet, with documentation, at <http://rude.sourceforge.net/>. Link checked 19th December 2004.
- [5-17] See New Skies web-site at www.newskies.com. Link checked 19th December 2004

6 Experimental results: the use of trace packets to measure user packet delay and loss

6.1 Introduction

In this chapter, results are presented from simulation models and from the test network, both of which are described in chapter 5. The chapter is laid out as indicated in figure 6-1.

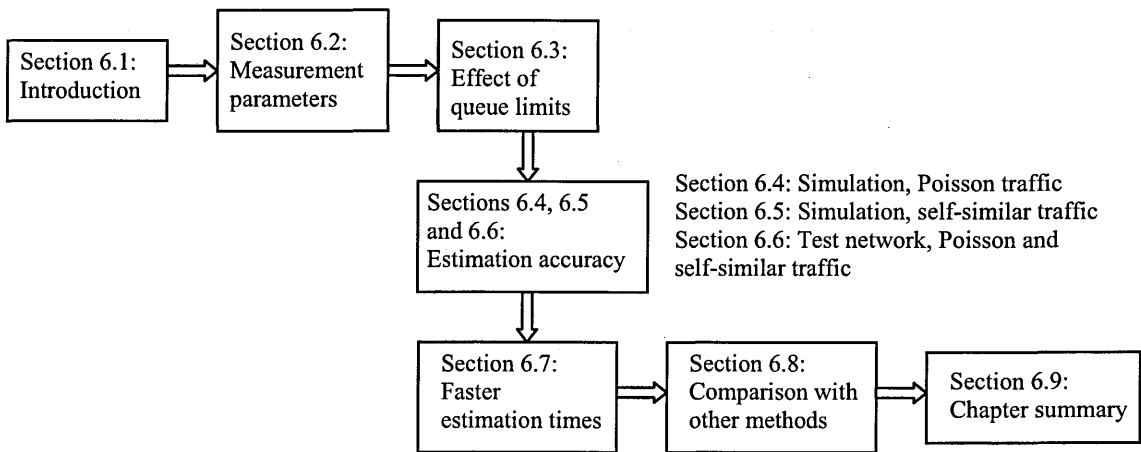


Figure 6-1 Layout of the chapter

Following the introduction, section 6.2 discusses the measurement parameters used during the experiment. Section 6.3 shows the impact of a queue limit on Poisson and self-similar traffic flows. Sections 6.4 and 6.5 contain results from simulations for Poisson and self-similar traffic sources respectively; these are split into two sections because of the large quantity of material. Section 6.6 contains results from the test network for both Poisson and self-similar sources. Section 6.7 contains results from a measurement designed to estimate the minimum time required to detect step changes in traffic load. Comparison of the trace packet estimation method with other methods such as direct sampling and SNMP polling is described in section 6.8. Finally, section 6.9 summarises the key results from the chapter.

The results in this chapter show the relationship between intensity of trace packets and the accuracy to which they provide estimates of delay and loss suffered by user packets in queues. The findings are basically that the estimation accuracy depends upon user packet length, network load and trace packet intensity. Generally, for any given trace packet intensity, greater accuracy is obtained with shorter user packets and higher loads. In order to clearly show this result, a novel approach is used in the presentation of results whereby the accuracy of packet delay estimation is plotted against trace packet variance.

As in chapter 5, a single queue is used to represent the network, justified on the basis that the satellite link is likely to be the dominant constriction in the path, so that there is one dominant queue, positioned just ahead of the satellite link.

A live GEO satellite link was available for a short period of time (2 days) as described in chapter 5 and was used to repeat the estimation accuracy results for Poisson traffic in section 6.6.1. The link added an expected fixed delay of around 306ms, but did not add perceptibly to the delay variation or to the packet loss, since no fades were encountered during the test period.

6.2 Measurement parameters

In an experiment of this kind, there are many parameters that can be varied. Examples are:

- user packet generation pattern,
- trace packet generation pattern,
- satellite link bandwidth and delay,
- measurement times,
- queue type and length.

It was considered necessary to limit the number of parameters varied during the tests in order to complete measurements in a reasonable amount of time, consistent with collecting sufficient results to test the hypothesis that trace packets can be used, with advantage on satellite links, as estimators of packet delay and loss. This section discusses the above list of parameters and the rationale for choosing them.

6.2.1 User packet generation patterns

Two generation patterns are used for user packets, these are Poisson and self-similar. Packet sources were constructed on both the simulation tool and the test network, with the ability to generate user packets of any length. The packets lengths actually used were limited to three: 64, 576 and 1500 bytes to represent those lengths which, according to several reports (e.g. Claffy [6-1]) are common on the Internet. It is suggested that these packet sizes come about because of the dominant types of service run over the Internet and can be roughly correlated with the service types indicated in table 4-2 as follows:

- < 64 byte packets tend to originate from TCP acknowledgements and applications where low packet fill delay is important, such as VoIP,
- 576 byte packets tend to originate from TCP sessions using segment lengths of 536 bytes,
- 1500 byte packets originate from LANs during FTP and other bulk data transfer where the edge routers perform fragmentation.

Most of the experiments were performed with 576 byte user packets, since this length carries the highest byte volume of traffic on the Internet. However one measurement (of delay estimation with a Poisson source) is repeated also for 64 and 1500 bytes in order to obtain a view of sensitivity to packet length. Table 6-1 shows which packet lengths were used against source generator type (Poisson / self-similar) and simulation / test network.

		User packet length (bytes)		
		64	576	1500
Simulation	Poisson	Yes	Yes	Yes
	Self-similar	No	Yes	No
Test network	Poisson	No	Yes	No
	Self-similar	No	Yes	No

Table 6-1 Packet lengths used in the experiments

The load on the queue ρ (in Erlangs) at a satellite earth station is a function of the packet length, the mean arrival rate and the satellite link rate. The user packet arrival rate is chosen to load the queue within the range $0.1 \leq \rho \leq 0.95$ for Poisson arrivals and $0.1 \leq \rho \leq 0.7$ for self-similar. The queue is not loaded beyond $\rho = 0.7$ for the self-similar source because beyond this the packet loss becomes unreasonably high for practical router queue

lengths, as will be shown later. The degree of self-similarity used throughout is represented by Hurst parameter $H = 0.75$ as described in chapter 5. There are occasions where $H = 0.95$ is mentioned in this chapter for purposes of comparison; in fact results were obtained for both values of H , but the results for $H = 0.75$ are used since this is more representative of traffic on the Internet.

6.2.2 Trace packet generation

The trace packet generation pattern is held constant, i.e. constant interval between arrivals. A few simulations were run with Poisson and self-similar trace packet generation patterns because it was thought early on that the accuracy of estimations might be higher if the trace packet generation pattern is the same as the user pattern. However the results of these were inconclusive, so a constant generation rate was adopted for trace packets to simplify the processing at the remote end of the satellite link: any delay variance detected there is then due to congestion in the queue. The trace packet intensity is typically varied between <1% to 10% of user packet mean rate. The length is fixed throughout at 64 bytes, considered to be as short as practicable while leaving enough room for a 20 byte header, a sequence number and a time-stamp.

6.2.3 Satellite link rate and delay

The satellite link service rate is fixed at 128 kbit/s throughout, since this is a typical average bandwidth made available to a single user or small network of users. There is no reason to suspect that changing the link rate will affect the findings, since the results relate to queue behaviour rather than the link rate. Since the link rate is held constant, the throughput in terms of user packets/second depends upon the packet length, as shown in table 6-2.

User packet length (bytes)	Throughput (packets / second) at 128 kbit/s
64	250
576	27.78
1500	10.66

Table 6-2 Relationship of packet length to packet throughput

Regarding the transmission delay, a practical GEO satellite link will add typically 300 ms for each hop. As a transparent satellite is assumed, the delay will be at the physical layer and will have zero short-term¹ variation. The satellite delay is ignored in the presentation of results in this chapter, since it will not impact upon the accuracy of estimations and it will vary slightly for different links depending upon the locations of the satellite and earth stations. For the time-series plots of delay, it is necessary to add a fixed delay of 300ms to get a typical end-to-end delay figure. However the satellite delay is discussed further in chapter 7 since it adds delay to time taken to detect changes in load on the network and must be added to the delay budget for management actions.

6.2.4 Measurement times

There are three classes of measurements described in this chapter:

¹ There is no delay variation over the time-scales considered in this study for queuing and measurement delays (say 10s to 100s of seconds). There is a small long-term delay variation caused by Doppler shifts which are over a time-scale of several hours.

- Simulations where the results are allowed to converge. These are run with simulation times of between 3000 - 20,000 seconds, depending upon user packet length and queue load. Generally, the higher the load, the longer is needed for convergence. These times were found by trial and error, firstly by carefully checking the early results by doubling the simulation duration times until there was no significant variation in the results ($< 1\%$) and secondly by randomly checking later results. This class of measurement is used in sections 6.3, 6.4 and 6.5
- On the test network, results were not always convergent due to an upper limit on the length of the packet capture file of 64,000 packets, imposed by the test equipment. This corresponds to a measurement time of around 2,560 seconds for $\rho = 0.9$, compared to the 20,000 seconds that are easily achievable using the simulation². For this reason and other reasons³, there are differences between the results from the test network and from the simulation, but the results still show the same trend. This is relevant to the network test results in section 6.6.
- Simulations and network test results which were deliberately not allowed to converge, in order to estimate the minimum time required to detect load steps. These measurements were run for 1000 seconds and relate to section 6.7.

6.2.5 Queue type and length

As discussed in chapter 5, the queue type used in the experiment was FIFO, since this unconditionally maintains packet order. A queue length limit is imposed with tail discard, whereby packets arriving to find the queue full are discarded. There are two length limits used in the experiment, these are:

- 100 Mbit, which is beyond the length that would result in any packet discards and therefore is virtually infinite for the purposes of this experiment. This limit is available by simulation only.
- 215 kbit, which is a typical practical queue length at the output of a router, as measured in chapter 5.

In the OPNET simulation tool, the queue limit was implemented by adding some lines of C code to the FIFO queue library process (full details of the code are given in Appendix A). By default, the OPNET queue length is limited only by the available memory of the machine on which it is running.

6.3 Effect of limiting the queue length

The effect of a queue length limit will reduce delay mean and variance at the expense of some packet loss due to discarding. Since self-similar traffic is more bursty than Poisson traffic for a given load, it is expected that self-similar traffic will be affected to a greater degree by a queue limit.

6.3.1 The effect of a queue limit on Poisson traffic

Figures 6-2 and 6-3 respectively show time-series plots of system delay experienced by Poisson generated user packets loading a queue at $\rho = 0.95$, with the queue length set at 100 Mbit and then at 215 kbit. The effect of the lower length limit is to 'clip' the delay peaks.

² The test network measurement periods are elapsed time, whereas simulation time is usually much faster. Depending on model complexity, 20,000 seconds of simulation time took typically 100 - 1000 seconds of elapsed time.

³ Other reasons include delay jitter and timing inaccuracies on the test network, these are explained in chapter 5.

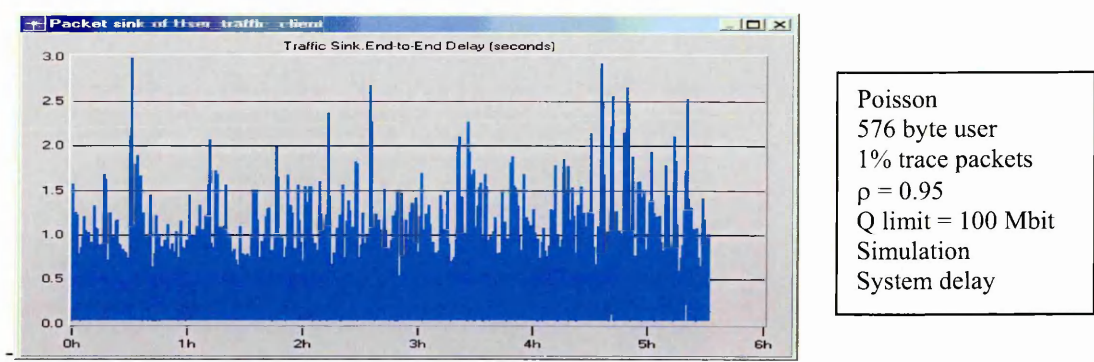


Figure 6-2 Time series of delay, 100 Mbit queue

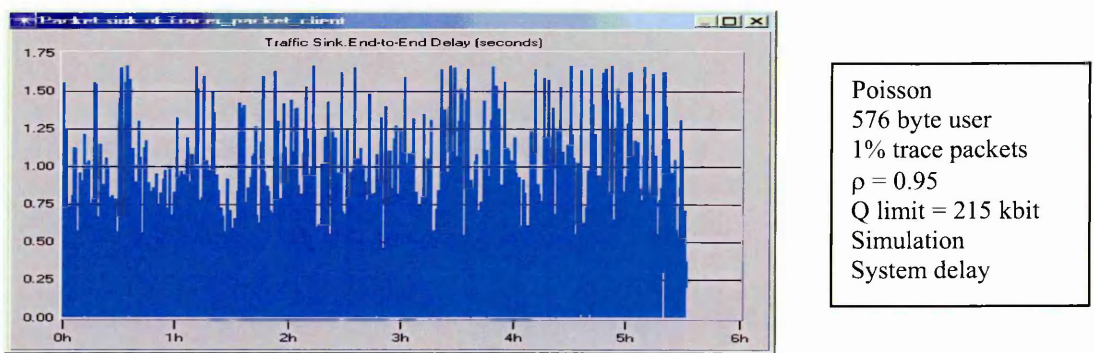


Figure 6-3 Time series of delay, 215 kbit queue

Figure 6-3 shows delay peaks clipped at 1.72 seconds which is the system delay, i.e. the time taken for the 576-byte packets to travel through the 215 kbit queue at 128 kbit/s when full (= 1.68 seconds), plus one service time of 0.04 seconds. Packet losses occur when the delay is clipped.

The delay and packet loss parameters for 100 Mbit and 215 kbit queue limits are shown in table 6-3, taken from the same data as figures 6-2 and 6-3.

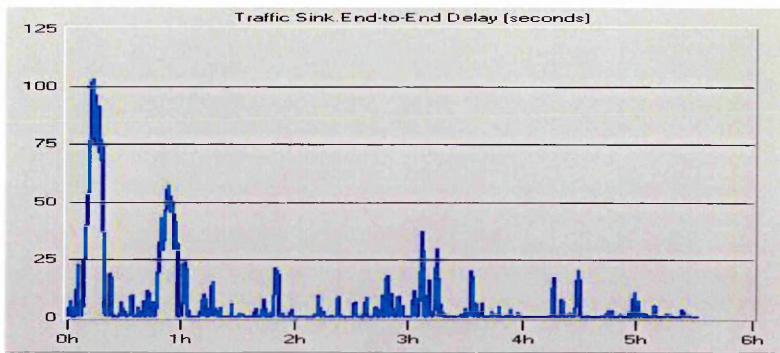
Queue limit	User packet delay mean / s	User packet delay variance / s ²	Packet loss / fraction of 528,636 packets sent
100 Mbit	0.3403	0.1195	0
215 kbit	0.3226	0.0942	4.8*10 ⁻⁴

Table 6-3 Effect of queue limit on delay and loss characteristics, $\rho = 0.95$, Poisson source

Table 6-3 shows that the delay mean and variance is reduced when the a practical limit is placed on the queue. There are no packet losses for $\rho < 0.9$.

6.3.2 The effect of a queue limit on self-similar traffic

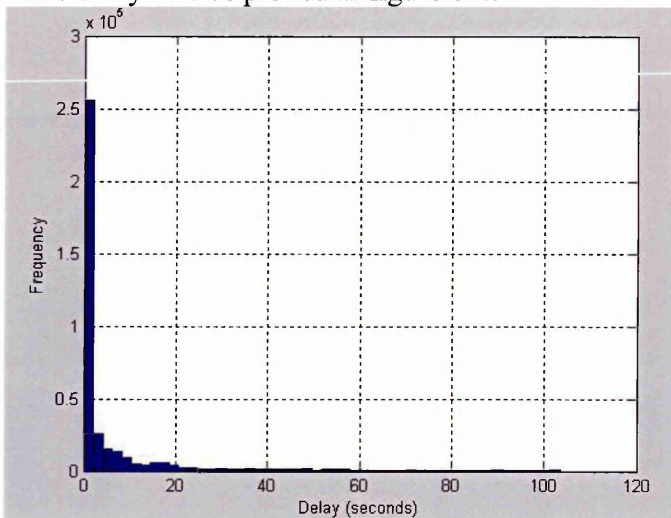
Figure 6-4 shows the time series for self-similar traffic ($H = 0.75$ throughout) with a 100 Mbit queue with $\rho = 0.7$. The x-axis is hours and the y-axis is seconds.



Self-similar ($H = 0.75$)
 576 byte user
 1% trace packets
 $\rho = 0.7$
 Q limit = 100 Mbit
 Simulation
 System delay

Figure 6-4 Time series of delay, 100 Mbit queue

The mean arrival rate λ_{tot} is $\rho * \mu = 19.44$ packets / second and the packet length is 576 bytes. Figure 6-4 shows that the queue length grows to over 1 Mbyte when the delay grows to over 100s about 15 minutes after simulation start. This traffic source is demonstrating a larger mean and variance than the Poisson case; in figure 6-4 the mean is 3.5 s and the variance is about 57 sec^2 . It can be seen graphically from figure 6-3 why any size buffer cannot be guaranteed to be large enough for self-similar traffic. With $H = 0.95$, the delay peaked at over 300 s. These large bursts do not occur with Poisson traffic, which is much better behaved. To give an idea of the delay distribution, figure 6-5 is a histogram of the same delay data as plotted in figure 6-4.



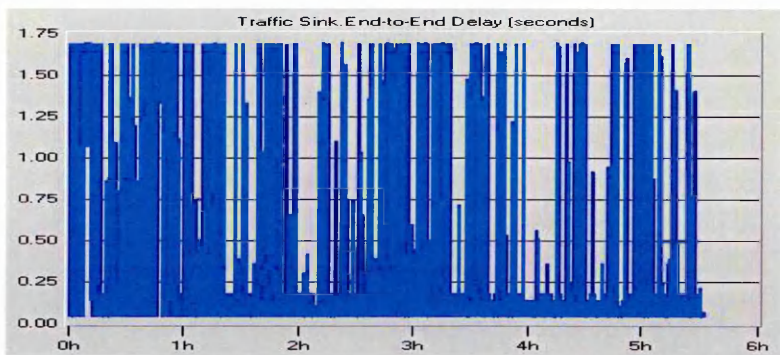
Self-similar ($H = 0.75$)
 576 byte user
 1% trace packets
 $\rho = 0.7$
 Q limit = 100 Mbit
 Simulation
 System delay

Figure 6-5 Delay histogram, 100 Mbit queue

Figure 6-5 shows that the distribution of system delay for self-similar traffic has a decaying nature for low delay values, but then it flattens out into a characteristic heavy tail to around 100 seconds.

A similar histogram constructed for $H = 0.95$ exhibited a tail that was flat to 300 seconds and it is reasonable to predict that the tail would continue to infinity if the source is exactly self-similar and the queue has infinite length. This tendency to infinite variance in exactly self-similar traffic (and hence towards infinite variance in queue delay) is described by Willinger as the ‘Noah effect’ [6-2].

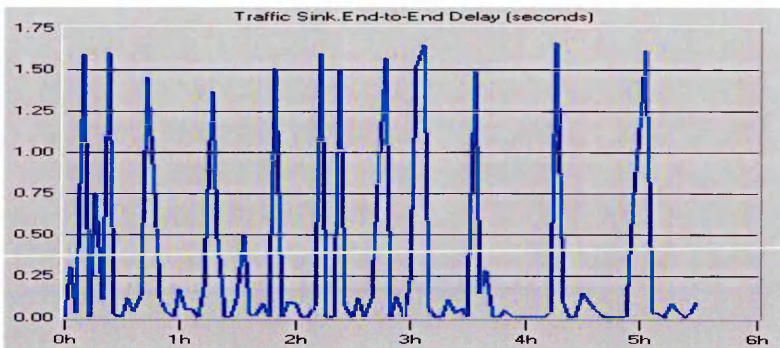
A queue limit of 215 kbit is now imposed on the self-similar source. Figure 6-6 and figure 6-7 show plots of the resulting delay time series of user packets and trace packets respectively. The trace packet intensity is 1% of the user packet mean.



Self-similar ($H = 0.75$)
 576 byte user
 $\rho = 0.7$
 Q limit = 215 kbit
 Simulation
 System delay

Figure 6-6 Time series of delay, user packets, 215 kbit queue

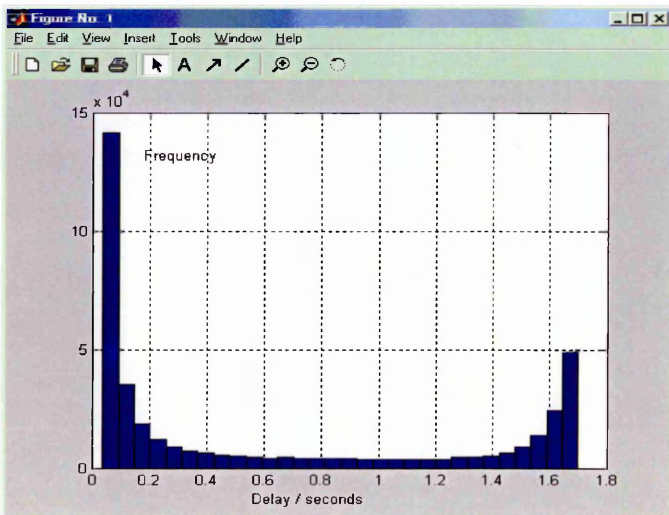
Comparing figure 6-6 with figure 6-3, the delay ‘clipping’ due to the queue limit is much more severe with self-similar traffic at $\rho = 0.7$ than it is for Poisson traffic at $\rho = 0.95$, because of the more bursty nature of self-similar traffic.



Self-Similar ($H = 0.75$)
 1% trace packets
 $\rho = 0.7$
 Q limit = 215 kbit
 Simulation
 System delay

Figure 6-7 Time series of delay, trace packets, 215 kbit queue

Figure 6-8 and figure 6-9 are delay histograms for user and trace packets respectively taken from the same data as figures 6-6 and 6-7. Note these delays are for the system.



Self-Similar ($H = 0.75$)
 576 byte user
 $\rho = 0.7$
 Q limit = 215 kbit
 Simulation
 System delay

Figure 6-8 Delay histogram, user packets, 215 kbit queue

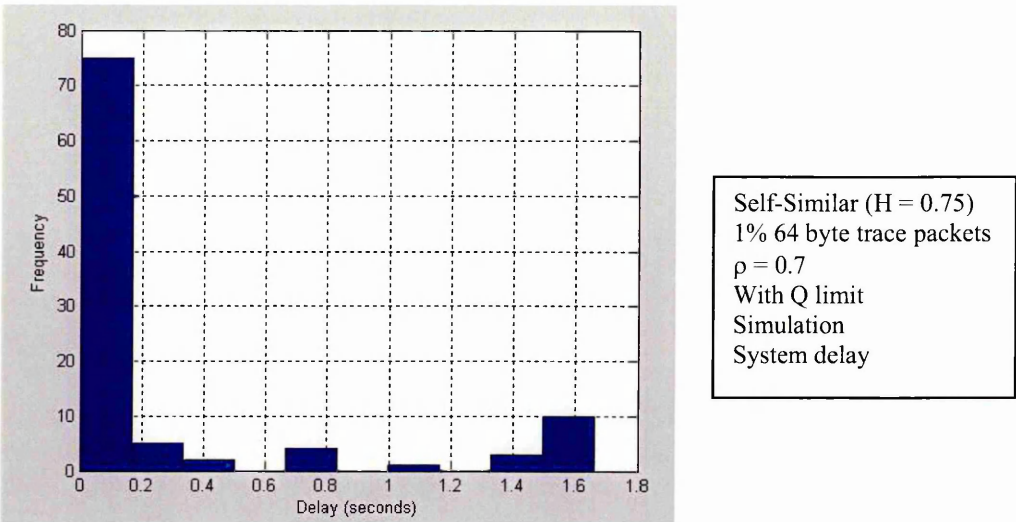


Figure 6-9 Delay histogram, trace packets, 215 kbit queue

The shape of the histograms in figures 6-8 and 6-9 shows bunching and clean cut-off of delay time as the queue limit is reached, producing double peaks. If the delay variation is specified using the percentile or interval method, there will be a higher number of packets above the upper bound than with Poisson traffic. The figures show that the user packet delay pdf shape can be estimated from the trace packet pdf; this point is important because the QoS standards recommend delay limits based on quantiles and percentiles.

The double-peak shapes in figure 6-8 and 6-9 are very similar to those derived from measurements of round-trip delay times encountered by ‘ping’ traffic performed by BT over a period of 6 months in a study for Eurescom [6-3]. In that report, the authors do not comment on the cause of the double peaks, but a private conversation revealed their opinion that the most likely cause is routine repeating of user behaviour. Bovy et al [6-4] have studied delay distributions on the Internet, including measurements done with ‘traceroute’, that also produce histograms with double peaks. They confess that they are uncertain of the origin of these peaks and suggest that they come from rapid switching of routes in the Internet. The work in the present study has shown that queue limits are a possible cause of double peaks measured by these other workers.

Regarding packet loss, self-similar traffic suffers more losses than Poisson because the queue limit is reached for a greater proportion of the time. The reason is that the bursts are larger with respect to the mean. As with the Poisson case, simulations were run where the packet losses were counted. Table 6-4 contains the summary results for delay and packet loss with 100 Mbit and 215 kbit queue lengths at $\rho = 0.7$.

Queue limit	User packet delay mean / s	User packet delay variance / s ²	Packet loss / fraction of 356216 packets sent
100 Mbit	3.5	57	0
215 kbit	0.32	0.27	0.018

Table 6-4 Effect of queue limit on user packet delay and loss characteristics, $\rho = 0.7$, self-similar source

Table 6-4 illustrates the effect of the 215 kbit queue limit in reducing the delay mean and variance. The mean is reduced by a factor of 10 and the variance is reduced by a factor of over 200, showing that the effect on the self-similar traffic is much greater than on Poisson traffic. Notice that the mean delay is virtually the same for Poisson and self-similar traffic in the limited queue ($= 0.32$ s).

The findings that limiting the queue length affects self-similar traffic more than Poisson traffic have also been confirmed by other workers. Bragg [6-5] has built self-similar traffic generators in OPNET and found that queue delays are longer and much more variable than for Poisson traffic; there is a plot in his paper that looks very similar to figure 6-4. Zhou and Sun have also performed simulations of self-similar traffic in queues [6-6] and their results are very similar, i.e. self-similar traffic suffers more delay in queues than Poisson traffic for the same network load, especially for higher loads. Also, with limited queues, they confirm that self-similar traffic suffers more packet loss than Poisson for the same queue size.

6.4 Simulation results of estimation accuracy, Poisson source

6.4.1 Effect on delay caused by adding trace packets

This section measures the effect on delay mean and variance of delay in the queue caused by adding trace packets to the user packet flow. A slight increase is expected due to the small additional load. Only the Poisson case with 100 Mbit queue is considered in this section, since the next section contains plots for both Poisson and self-similar traffic that allow comparisons to be made. A lower queue limit is not tried, since such a limit has been seen to reduce the delay mean and variance and hence the effect of trace packets is assumed to be less.

Adding trace packets to the user packet flow places a small additional load on the network, the magnitude of which depends upon the intensity of the trace packets added and their length as a function of the user packet length. For example, if the user packets and trace packets are both 64 bytes, then adding 1% trace packets will add 1% directly to queue load, whereas the load increase is much less if the user packets are 1500 bytes.

Although the increase in load due to trace packets will cause a small increase in delay, the effect when using 576 byte user packets is not significant until the queue load ρ approaches 1. To illustrate this, figure 6-10 shows the queue delay mean and variance of 576-byte user packets as the queue load is varied, with packets from a Poisson source.

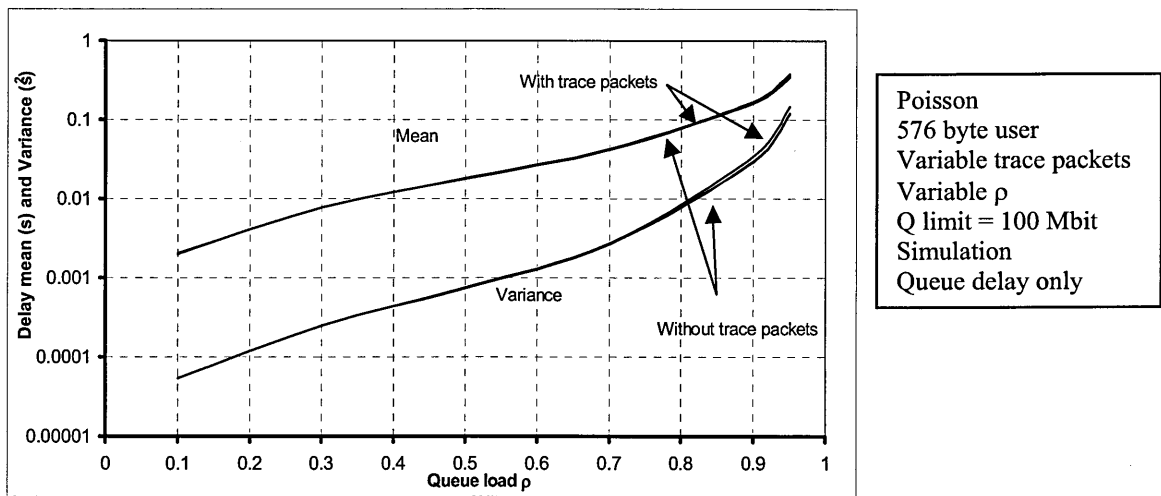


Figure 6-10 Effect on user packet delay mean and variance caused by adding trace packets

There are 10 plots in figure 6-10, 5 for the mean and 5 for the variance, each with a different intensity of trace packets present: 0, 1, 2, 5 and 10% of the user packet mean. The plots are virtually on top of each other, except for a slight divergence at higher values of ρ . This indicates that the presence of trace packets does not significantly affect the delay mean or variance until ρ approaches 1. Since there is sometimes some doubt about the use of log-scale plots for such comparisons, table 6-5 shows the actual values of the effect of trace packets at $\rho = 0.95$, which gives the worst-case effect.

Trace packet intensity	User packet delay mean (s)	User packet delay variance (s ²)
0%	0.340	0.119
1%	0.344 (+1.2%)	0.121 (+1.7%)
2%	0.345 (+1.5%)	0.121 (+1.7%)
5%	0.358 (+5.3%)	0.133 (+11.8%)
10%	0.378 (+11.2%)	0.147 (+23.5%)

Table 6-5 The effect of trace packets on user packet delay for $\rho = 0.95$

Table 6-5 shows that increases in the delay mean approximately track the trace packet intensity but the variance is affected more because of the higher slope of the variance curve at $\rho = 0.95$. The main result taken from table 6-5 is that, for this model, if the effect on the delay mean and variance is to be kept below about 2% at the highest queue loads, the maximum trace packet intensity should be restricted also to 2%.

6.4.2 Absolute measurements of error

Results of estimation error of user packet mean delay in the queue are shown in figure 6-11 for Poisson traffic. This shows how trace packet mean delay tracks user packet mean delay for various queue loads against number of trace packets used in the estimation.

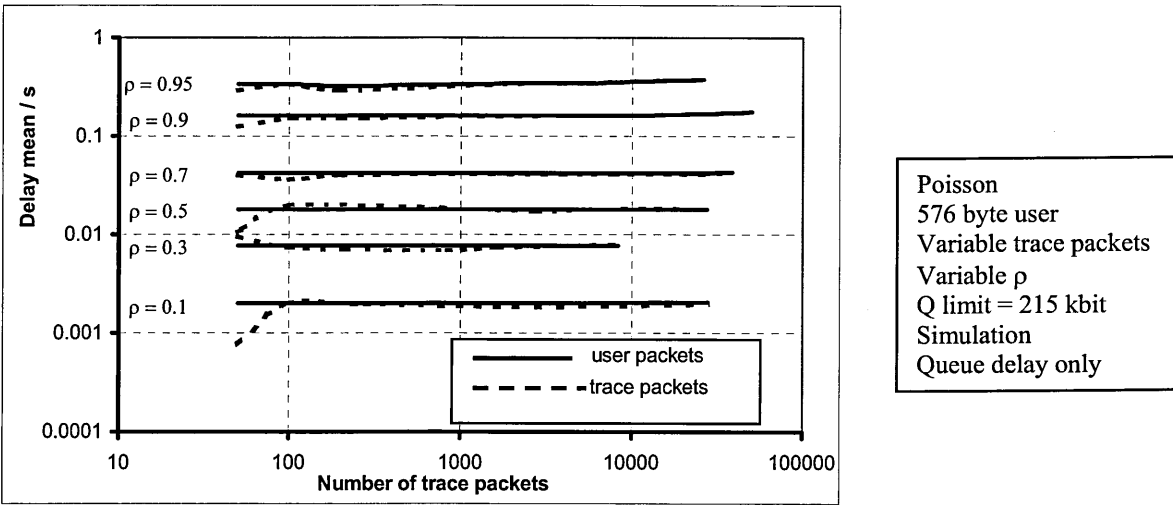


Figure 6-11 Delay mean against number of trace packets used for estimation

Figure 6-11 illustrates that:

- the estimation accuracy increases with the number of trace packets, as expected,
- the estimation goes above and below the true value of mean delay, which is expected from this unbiased estimation method. For low numbers of trace packets under-estimation appears to be more likely,
- the mean delay of user packets does not increase visibly as a result of adding trace packets until $\rho > 0.9$ where the plots rise slightly for greater numbers of user packets.

Figure 6-12 shows the results for estimation of the variance. The same findings apply as for the estimation of mean, except that the errors are greater; this is because the variance curve is steeper than the mean curve against ρ , confirming the indication in figure 6-10 and table 6-5.

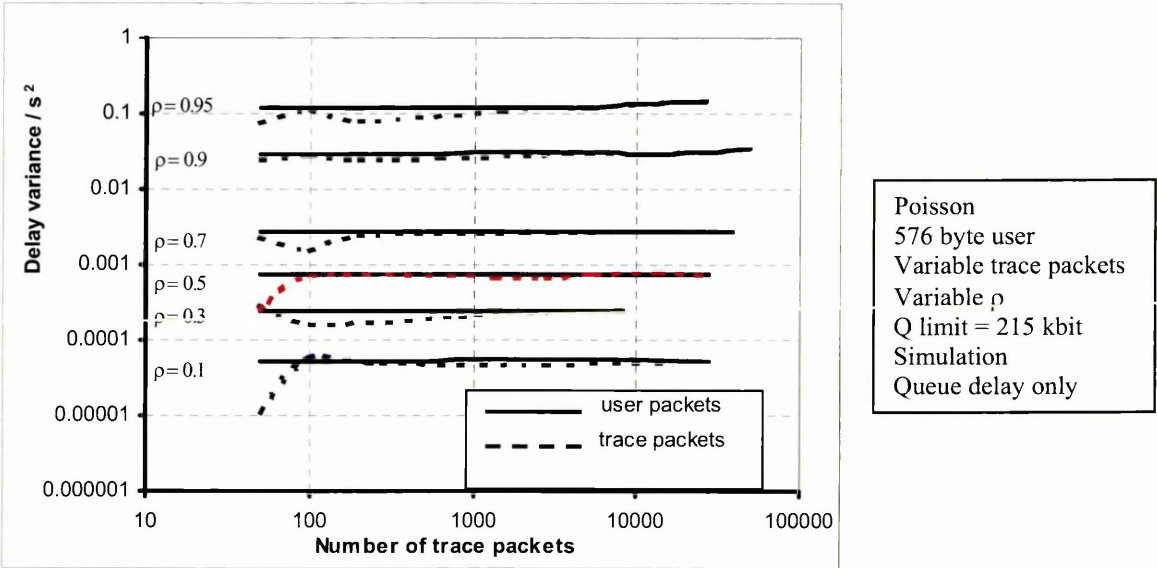


Figure 6-12 Delay variance against number of trace packets used for estimation

The y-axes in figures 6-11 and 6-12 are the absolute values of delay mean and variance and the x-axes are the number of trace packets used. However it was decided to investigate other ways of presenting the results that may be more useful. It was noticed while running these simulations that the estimation accuracy increases with queue load for a given number of trace packets, which these plots do not show particularly clearly. This can be seen in some cases, for instance the plot lines diverge more at $\rho = 0.1$ than at $\rho = 0.5$. One alternative presentation is to overlay probability distributions (pdfs) of the user and trace packet delays.

Figure 6-13 shows a delay pdf formed from 64,000 user packets and just 64 trace packets, showing that a reasonable estimation of the pdf can be obtained with a 1:1000 trace packet intensity, with approximately equal under- and over-estimates of the shape.

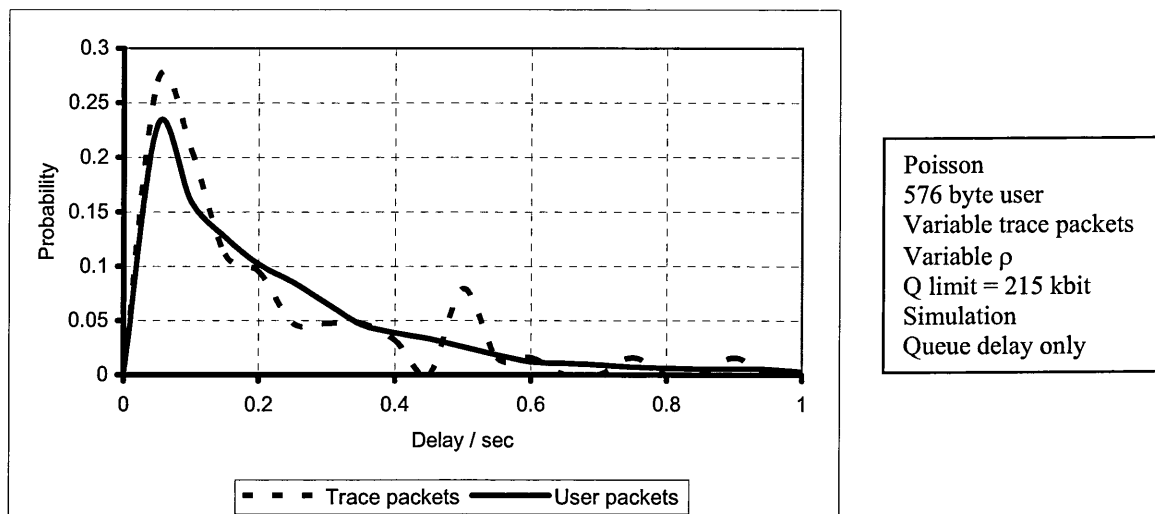


Figure 6-13 Delay pdf of user and trace packets at load $\rho = 0.95$

6.4.3 Plotting of results against trace packet variance

To try to find a presentation method that clearly shows how the estimation accuracy increases with load, while at the same time showing the sensitivity to number of trace packets, it was decided to experiment with re-plotting the system delay data as a percentage estimation error against trace packet variance. This novel approach has the following advantages:

- the variance can easily be measured at the remote end of satellite links, since the packets are inserted on a regular time grid (subject to resolving of clock skews),
- the variance can be used as an input to a management decision process about which measurement method to use, e.g. trace packets or direct sampling of user packets (more on this later),
- the variance is an indication of congestion and can be used to set the level of trace packet intensity,
- the variance can indicate to the management process which mode it should be in, i.e. 'normal', 'intermediate' or 'damage limitation' as defined in table 4-1 in chapter 4.

The other changes to the presentation results made at this stage are:

- the percentage error is used instead of the absolute values of mean and variance on the y-axis,
- the trace packet intensity is varied according to the following steps: 1, 2, 5 and 10% instead of absolute numbers of trace packets on the x-axis.

The same data used to plot figures 6-11 and 6-12 are re-plotted as figures 6-14 and 6-15 with these parameter changes, showing the mean and variance error respectively. In the figures, the loci of the lines follow the number of trace packets used.

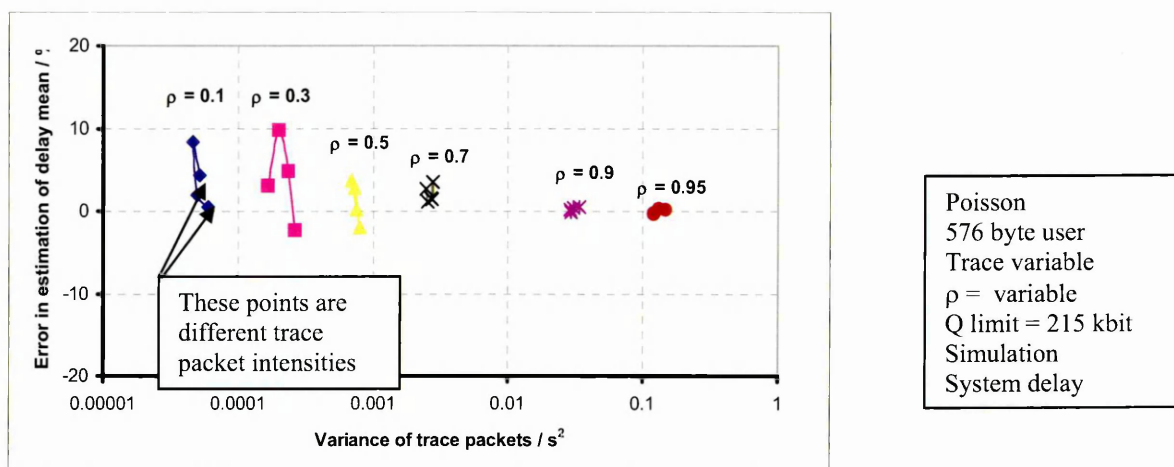


Figure 6-14 Percentage error in estimation of user packet delay mean against trace packet variance

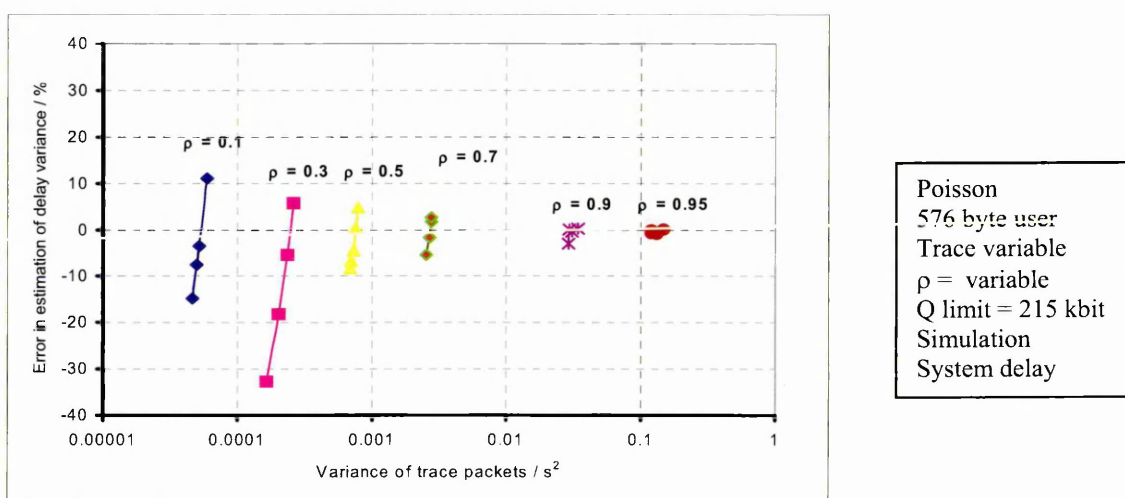


Figure 6-15 Percentage error in estimation of user packet delay variance against trace packet variance

Figures 6-14 and 6-15 show that:

- the estimation accuracy for mean and variance does indeed improve as a function of the trace packet variance, which itself increases as a function of queue load,
- for the mean, the estimations approach the true value from over-estimations, while from the variance the true value is approached from both over- and under-estimations,
- there is less sensitivity to the trace packet intensity used as the load increases, as displayed in the figures by reduction in the enclosed areas of each cluster. This applies more to the variance than to the mean (note change of y-axis scale).

What this plotting method does not show clearly is which end of the clusters correspond to the least or greatest intensity of trace packets since some of the lines double-back on themselves. However, in every case the most accurate estimate corresponds to the highest intensity (10%) and the continuous line goes sequentially through the other intensities down to 1%.

It is worth repeating that these results are of potentially great interest for satellite links. They indicate that the in-band measurement traffic, when represented by trace packets, can

be progressively reduced as congestion increases while maintaining a given measurement accuracy. This compares with the need to increase measurement overhead traffic to maintain accuracy with increasing congestion when direct sampling of user packets is employed (some calculations are done on this later in the chapter). Moreover, the congestion itself can be estimated from the variance of the trace packet delays, so that trace packet variance can be used to automatically set the trace packet intensity for the required accuracy.

6.4.4 Different length user packets: 64 and 1500 bytes

The results so far in this section have been for 576 byte user packets. Bearing in mind that 64 byte and 1500 byte packets are also common on the Internet, the simulations are repeated with the user packet length changed to 64 and then to 1500 bytes (but with trace packets still at 64 bytes). The results for 64 byte user packets are shown in figures 6-16 and 6-17, and those for 1500 byte user packets are shown in figures 6-18 and 6-19. For these plots, trace packet intensities were varied over 0.5, 1, 2, 5, and 10%.

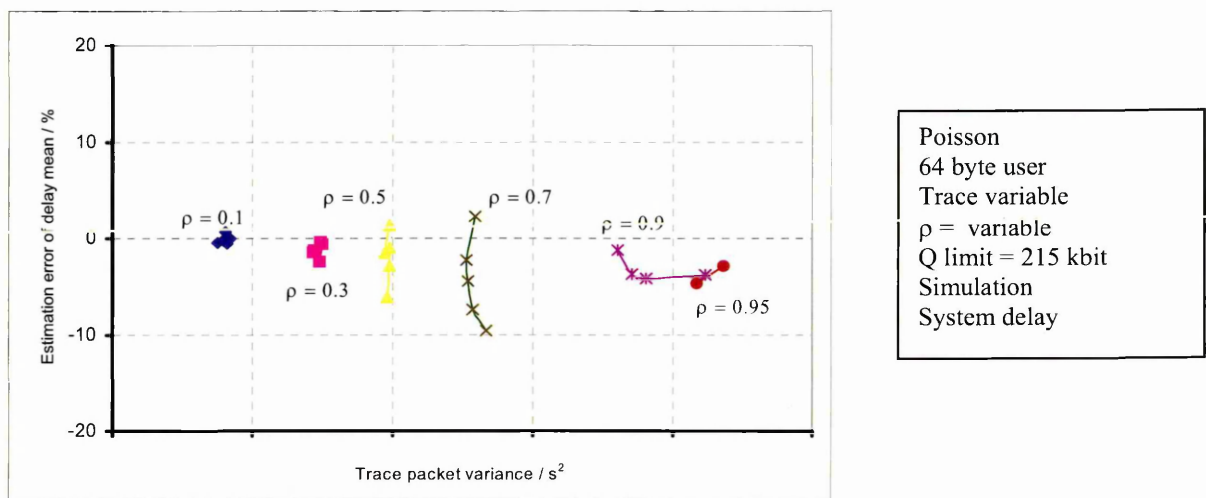


Figure 6-16 Error in estimation of user packet delay mean, 64 byte user packets

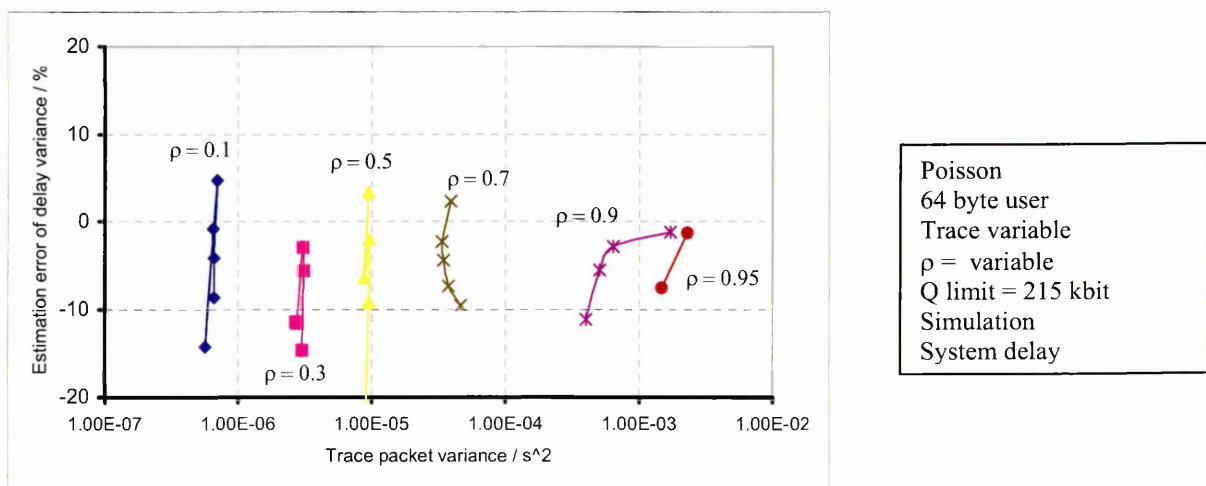


Figure 6-17 Error in estimation of user packet delay variance, 64 byte user packets

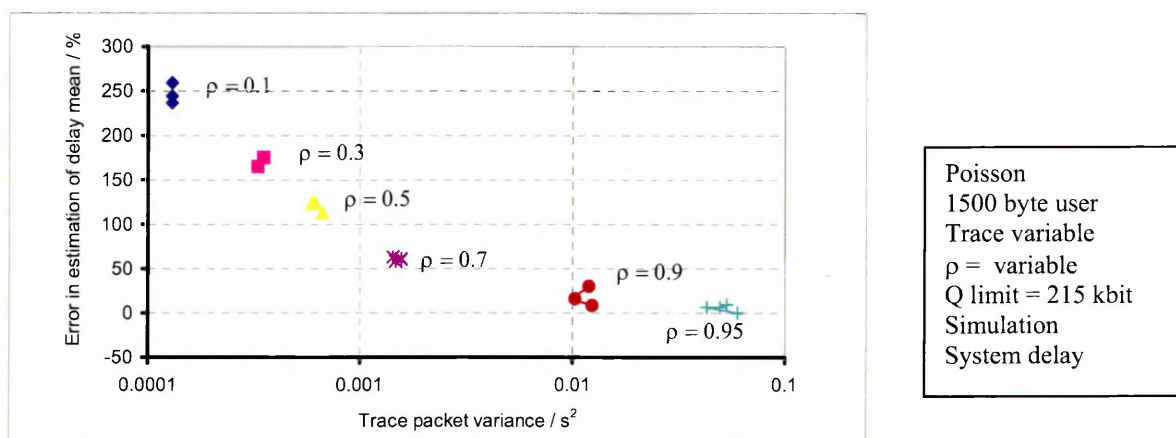


Figure 6-18 Error in estimation of user packet delay mean, 1500 byte user packets

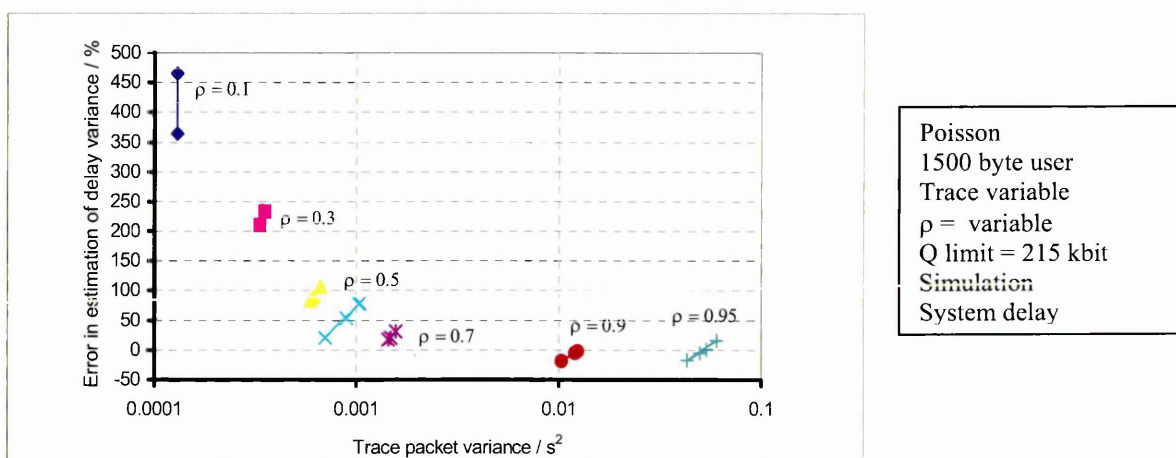


Figure 6-19 Error in estimation of user packet delay variance, 1500 byte user packets

Figures 6-16 and 6-17 show that, when the user packets are the same length as the trace packets (=64 bytes), the accuracy of the estimate remains approximately constant as the load increases. The changes in area traced out by the clustering indicates that the sensitivity to trace packet intensity increases as the load increases to about $\rho = 0.7$ but then reduces as the load is increased further. The estimations approach the true values from under- and over-estimations for both the mean and the variance.

In contrast, figures 6-18 and 6-19 show that, for 1500 byte user packets, the decrease in accuracy with decreasing load is even more marked than in the 576-byte case. The cluster areas stay approximately equal, which means that there is a constant sensitivity to trace packet intensity. The estimations approach the true value from over-estimations for the mean and the variance.

Table 6-6 summarises these results. In all cases the estimation of mean is more accurate than the estimation of variance.

User packet length (bytes)	Accuracy of estimation	Sensitivity to trace packet intensity
64	Independent of ρ , but within +/- 15%	Increases up to $\rho = 0.7$, then decreases for $\rho > 0.7$
576	Decreases with low ρ , over-estimates mean. Within -33 to +10%.	Decreases with ρ
1500	Decreases with low ρ , over-estimates mean and variance. Within -30 to over 400%.	Independent of ρ

Table 6-6 Summary of results of delay estimation accuracy with different user packet lengths, Poisson, simulation

The reasons for this interesting behaviour could be that:

- longer user packets suffer more delay and delay variation in queues, since the discrete jumps in queue length are larger when a new packet arrives,
- as the queue becomes more loaded, the delay / load curve becomes steeper and more trace packets wait behind user packets in the queue and hence contribute to the accuracy of the estimation.

These points are discussed more fully at the end of the chapter.

6.4.5 Estimation of packet loss

Section 6.3.1 showed that the effect of limiting the queue length to 215 kbit caused packet loss only for $\rho > 0.9$ with Poisson traffic. At lower loads, the queue was never full during 20,000 seconds of simulation time so that there were no losses. As the effect of the queue limit is significant only for $\rho > 0.9$, it is not efficient to show it by way of plots against ρ . Therefore table 6-7 has been constructed to show the accuracy in loss estimation for $\rho = 0.95$, for various values of trace packet intensity from 1 to 10%. The estimation error is calculated using:

$$[(\text{Trace packets lost (fraction)} - \text{User packets lost (fraction)}) / \text{User packets lost (fraction)}] * 100\%.$$

Trace intensity (%)	Trace packets sent	Trace packets received	Trace packets lost (fraction)	User packets sent	User packets received	User packets lost (fraction)	Estimation error (%)
1	5279	5276	$5.7 * 10^{-4}$	528636	528382	$4.8 * 10^{-4}$	+19%
2	10556	10551	$4.7 * 10^{-4}$	528737	528460	$5.2 * 10^{-4}$	-10%
5	26386	26371	$5.7 * 10^{-4}$	528467	528131	$6.3 * 10^{-4}$	-9%
10	52770	52722	$9.1 * 10^{-4}$	528438	527937	$9.6 * 10^{-4}$	-5%

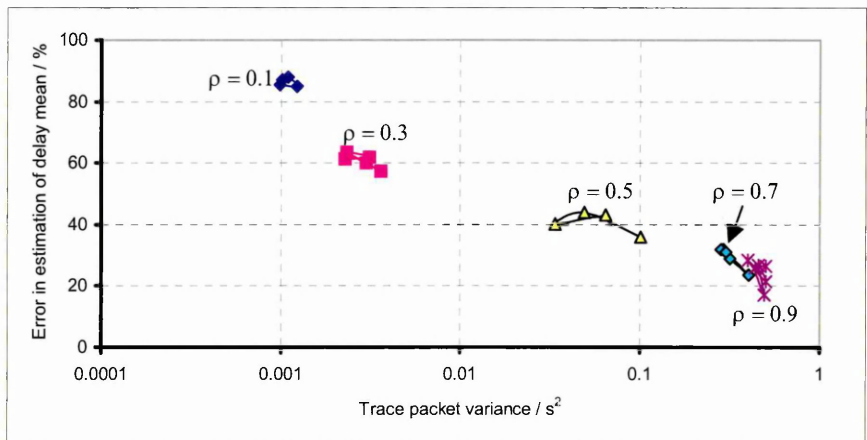
Table 6-7 Packet loss estimation error, Poisson traffic, $\rho = 0.95$, simulation

Table 6-7 shows that (as expected) the estimate of packet loss improves with more trace packets. Notice that the trace packet estimations generally under-estimate the packet loss (the 1% figure can be discounted from this comment since it is taken from a very small number of trace packets lost (=3), if one less had been lost then this too would have under-estimated). Later, when these results are compared with estimations of packet loss for self-similar traffic, the degree of under-estimation becomes worse.

6.5 Simulation results of estimation accuracy, self-similar source

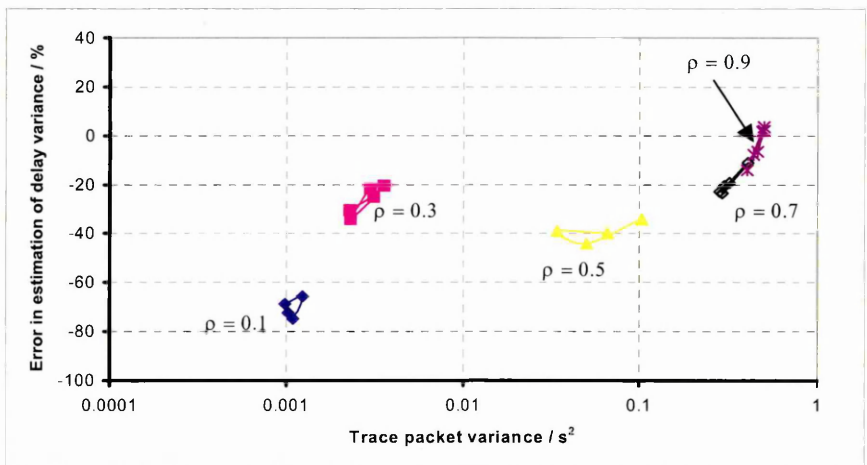
6.5.1 Estimation of delay

The estimation error of user packet delay mean and delay variance for self-similar traffic against trace packet variance is shown in figures 6-20 and 6-21 respectively (compare with figures 6-14 and 6-15 for the Poisson case). A queue limit of 215 kbit has been imposed and $H = 0.75$.



Self-similar, $H = 0.75$
576 byte user
Trace variable
 ρ = variable
Q limit = 215 kbit
Simulation
System delay

Figure 6-20 Error in estimation of user packet delay mean



Self-similar
576 byte user
Trace variable
 ρ = variable
Q limit = 215 kbit
Simulation
System delay

Figure 6-21 Error in estimation of user packet delay variance

Figures 6-20 and 6-21 show that:

- the accuracy improves with queue load as with Poisson traffic (except perhaps for $\rho = 0.3$ to 0.5 in figure 6-21),
- for the variance, the estimates approach the true value from under-estimations, while for the mean the estimates approach the true value from over-estimations. This is the same for both Poisson and self-similar sources but the effect is more pronounced in the self-similar case,
- there is little increase in variance between $\rho = 0.7$ and 0.9 , because the queue is already discarding significant numbers of packets at $\rho = 0.7$ and not many more are discarded at $\rho = 0.9$,
- the spread of measurements with different trace packet intensities is roughly equal whatever the load, as shown by the length of the plot lines staying roughly constant. The most accurate estimations are always from the greater number of trace packets.

- the estimation accuracy is not as good as in the Poisson case, with 20% being typical for $\rho \geq 0.7$.

6.5.2 Estimation of packet loss

The estimate made of user packet loss by trace packet loss in the self-similar case is shown in table 6-8.

Trace intensity (%)	Trace packets sent	Trace packets received	Trace packets lost (fraction)	User packets sent	User packets received	User packets lost (fraction)	Estimation error (%)
1	4000	3954	0.011	397437	376814	0.052	-78%
2	8000	7816	0.023	470935	432268	0.082	-72%
5	20000	19816	0.0092	404720	384589	0.05	-81%
10	40000	39632	0.0092	419833	396815	0.055	-83%

Table 6-8 Packet loss estimation error, self-similar traffic, $\rho = 0.7$, simulation

The trace packets under-estimate the user packet loss more severely than for the Poisson case (compare with table 6-7). It is interesting that, just as in the delay estimation, there is not very much sensitivity to trace packet intensity. It is reasonable to assume that estimates of self-similar traffic will not improve with measurement time or sample size as much as Poisson traffic, consistent with the slow drop in variance with increasing aggregation.

6.6 Test network results, Poisson and self-similar sources

Experiments were performed on the test network described in chapter 5 to assess the effectiveness of trace packets as estimators of packet delay and loss in a practical network environment. Most of the measurements are repeated from sections 6.4 and 6.5, but with fewer parameters as follows:

- User packet length fixed at 576 bytes and trace packet length fixed at 64 bytes,
- Load values are 0.5, 0.7 and 0.9 for the Poisson source and 0.3, 0.5 and 0.7 for the self-similar source.

6.6.1 Estimation of packet delay, Poisson traffic

Figures 6-22 and 6-23 were taken from the test network and from the live satellite link and show the practical realisation of the trace packet estimation method for the Poisson case. The trace packet intensity was varied over 1, 2, 5 and 10% of user packet mean and results were taken for $\rho = 0.5, 0.7$ and 0.9 . The plots show the accuracy of estimations for delay mean and variance against trace packet variance and should be compared with figures 6-14 and 6-15.

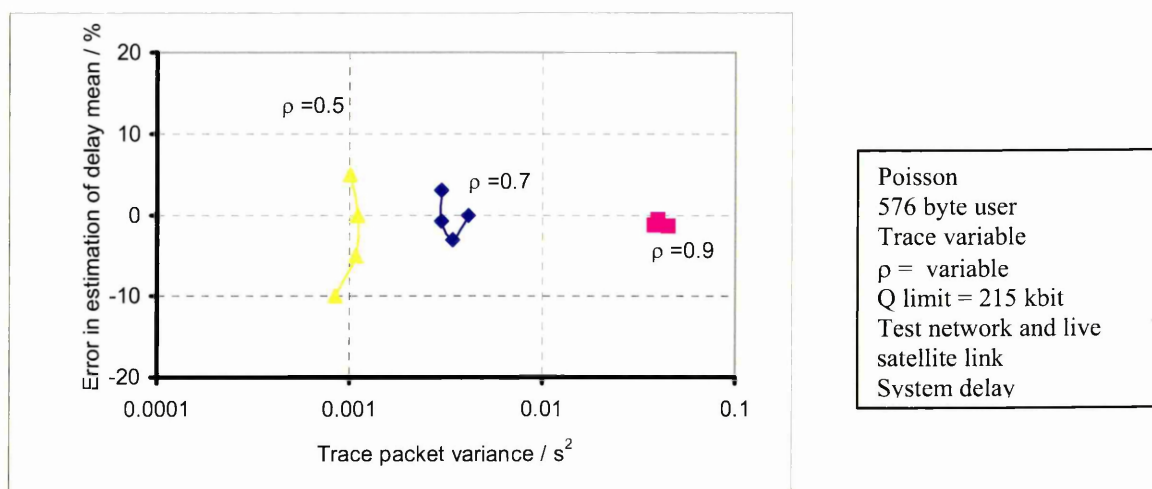


Figure 6-22 Error in estimation of user packet delay mean

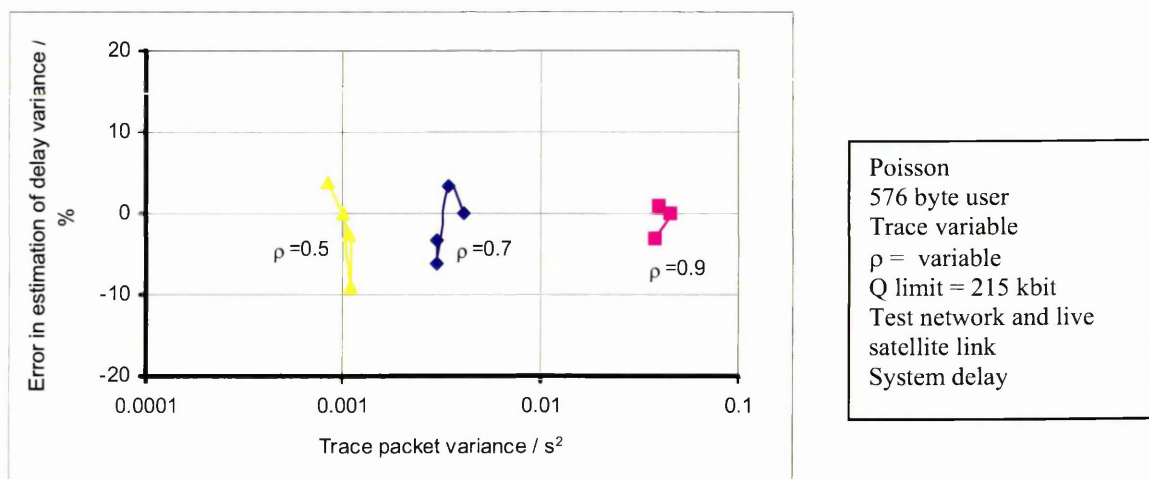


Figure 6-23 Error in estimation of user packet delay variance

Comparing the plots in figures 6-14 and 6-15 with 6-22 and 6-23, it can be seen that the OPNET simulation results are very close to the test network results. The mean value is approached from slight over estimates and the variance value is approached from estimations symmetrically about zero error in both the simulation and real network. In looking at these results it should be noted that:

- the simulation has perfect timing at both ends of the network, whereas the clocks at either end of the test network, although running at exactly the same speed (since they have a common reference), they carried an uncertainty of a few milliseconds in absolute time when logging packet arrivals,
- it was difficult to detect which packets the test network had discarded and so remove them from the end-to-end delay calculations. It was necessary to manually track the queue occupancy.

6.6.2 Estimation of packet loss, Poisson traffic

Table 6-9 shows the results for 1, 2, 5 and 10% trace packet intensities.

Trace intensity (%)	Trace packets sent	Trace packets received	Trace packets lost (fraction)	User packets sent	User packets received	User packets lost (fraction)	Estimation error (%)
1	590	590	0	46540	46518	$4.7 \cdot 10^{-4}$	n/a
2	1180	1179	$8.4 \cdot 10^{-4}$	46540	46518	$4.7 \cdot 10^{-4}$	+78
5	2950	2948	$6.8 \cdot 10^{-4}$	46540	46517	$4.9 \cdot 10^{-4}$	+39
10	5900	5897	$5.1 \cdot 10^{-4}$	46540	46516	$5.1 \cdot 10^{-4}$	0

Table 6-9 Packet loss estimation error, Poisson traffic, test network, $\rho = 0.95$

Table 6-9 shows that the number fraction of user packets lost is close to that of the simulation in table 6-7. There is a constant number of user packets sent because the test network traffic generator read through the same file each time. The number of trace packets lost was very low (maximum is 3) and so the level of confidence in the estimation is small. To improve this situation either a higher trace packet intensity or a longer measurement time would be needed. Despite the few trace packets lost, the fraction of trace packets lost is in broad agreement with the user packets.

6.6.3 Estimation of packet delay, self-similar traffic

The error in estimation of delay mean and variance is shown in figures 6-24 and 6-25 respectively for $\rho = 0.3, 0.5$ and 0.7 . These plots should be compared with the simulation results in figures 6-20 and 6-21. The estimation of packet loss is shown in table 6-10 which likewise should be compared with table 6-8.

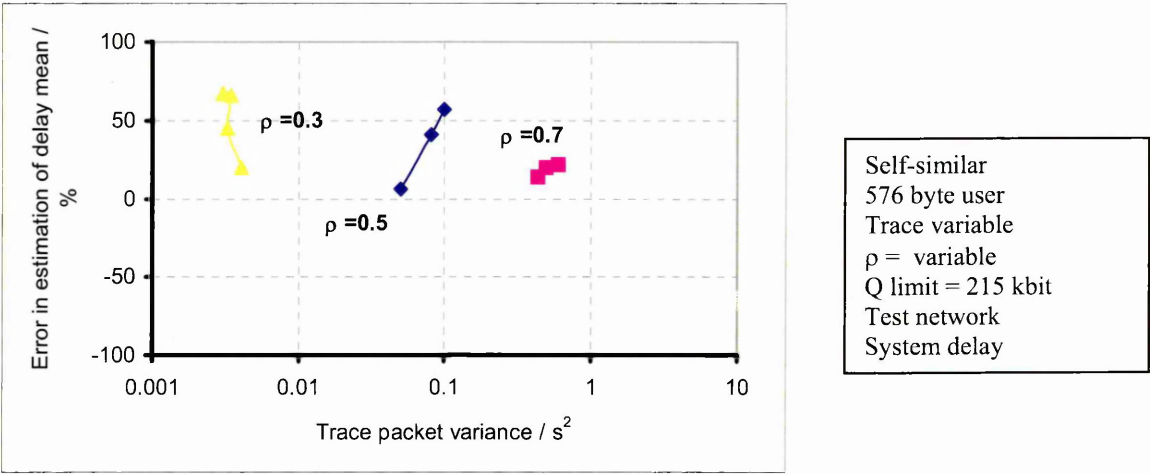


Figure 6-24 Error in estimation of user packet delay mean

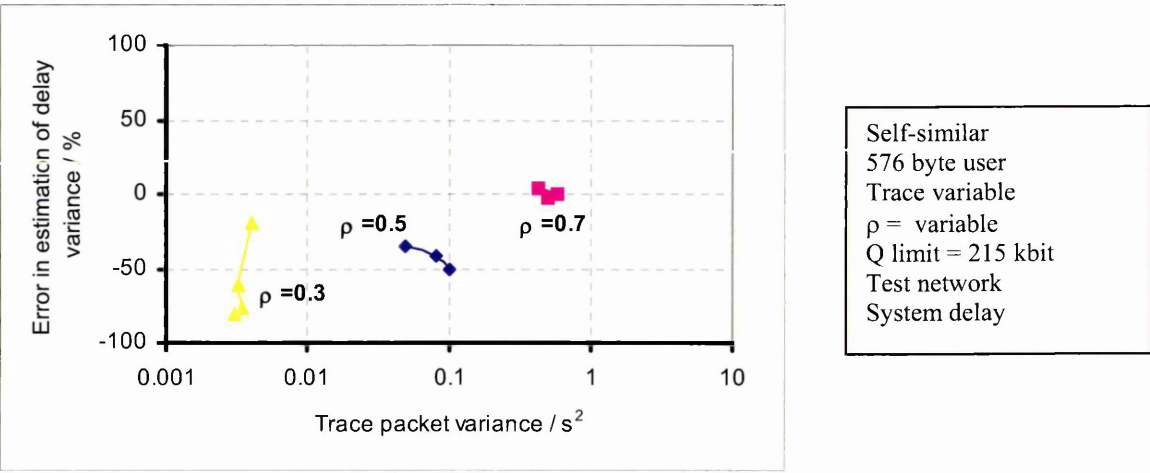


Figure 6-25 Error in estimation of user packet delay variance

Comparing figure 6-24 with 6-20 and figure 6-25 with 6-21, they are similar for $\rho = 0.3$ and 0.5 but the test network gives more accurate estimations than the simulations for $\rho = 0.7$ for the variance.

6.6.4 Estimation of packet loss, self-similar traffic

Table 6-10 shows the results from the test network for packet loss estimation.

Trace intensity (%)	Trace packets sent	Trace packets received	Trace packets lost (fraction)	User packets sent	User packets received	User packets lost (fraction)	Estimation error (%)
1	270	267	0.01	38580	36651	0.05	-80
2	530	522	0.015	38580	36636	0.05	-70
5	1340	1297	0.032	38580	36501	0.054	-41
10	2680	2584	0.036	38580	36317	0.059	-39

Table 6-10 Packet loss estimation error, self-similar traffic, test network, $\rho = 0.7$

The packet loss estimations in table 6-10 show that trace packets under-estimate the losses, just as with the simulation. From tables 6-7, 6-9 and 6-10 it does seem that the use of trace packets generally under-estimate user packet losses and the under-estimation is worse for self-similar traffic. A correction factor could be built into the loss estimation, but such a factor would probably depend on the degree of self-similarity which may not be known.

Despite the relatively poor accuracy, it is still argued that trace packets should be used to detect packet loss on satellite links, since the alternative is to tag and count user packets, which is costly in terms of processing power. The implication is that any trace packet losses should be reported to the OSS as soon as possible, since the loss of user packets will almost certainly be higher. An event reporting such a loss should be handled with high priority.

6.7 Investigation into faster estimation times

Having established the effectiveness of trace packets for estimating user packet delay characteristics over long estimation periods (i.e. hundreds to thousands of seconds), with higher queue loads, attention is now turned to a more practical matter of detecting changes in load in more realistic (i.e. much shorter) timescales. A measurement period of 1 minute is proposed for SLA reporting in the ITU-T standard Y.1541 [6-10]. Recall also from chapter 2 that 35 – 55 seconds interruption to TCP sessions causes them to time out. It is of interest to assess the implications of these shorter times. This section determines the minimum time required to detect a change in load so that a management process can react with a reasonable degree of confidence, within the context of the models used.

6.7.1 Response of the queue to steps in load

Figure 6-26 shows the time-series of delay from the system when handling user packets from a Poisson source when ρ is suddenly increased from 0.5 to 0.9. The user packets are 576 bytes. Figure 6-27 shows the result for self-similar traffic for a step in ρ from 0.3 to 0.7. As elsewhere in this study, the service rate $\mu = 27.78$ packets/s and the queue limit is 215kbit.

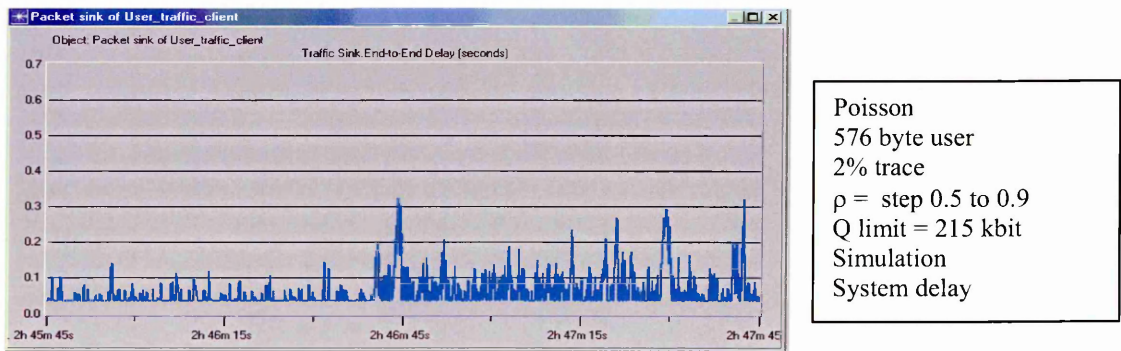


Figure 6-26. Response in delay for step increase of ρ from 0.5 to 0.9. User packets from Poisson source

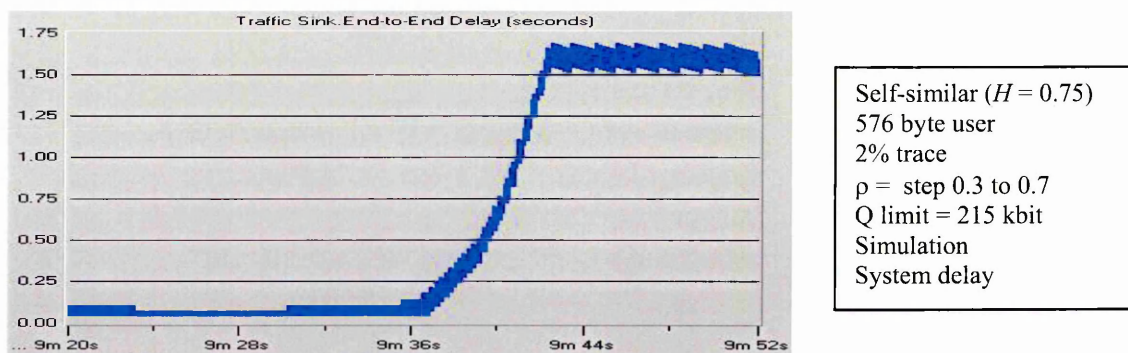


Figure 6-27. Response in delay for step increase of ρ from 0.3 to 0.7. User packets from self-similar source

Figure 6-26 and figure 6-27 show that the delay climbs over a few seconds after the step is applied, with Poisson traffic showing a much more modest increase in delay for the same size step in load. In the self-similar case, the queue overloads approximately 6 seconds after the step is applied, due to the high peak to mean ratio of the arrivals.

This implies that practical queue lengths give only a few seconds buffering after a step change in load. A requirement on the management system to react in less than around 6 seconds if packet losses are to be avoided in this particular system. In order to determine whether it is possible for trace packets to detect these changes in load in this time-scale, experiments were performed by simulation using OPNET and by measurements on the test network.

The following parameters were used:

- Simulation and test-network measurement time is fixed at 1000 seconds, with the step occurring at 500 seconds,
- Trace packet intensity = 2% (since this was found to alter the delay and variance of the delay by < 2%),
- Poisson traffic load steps of $\rho = 0.1$ to 0.5, then from 0.5 to 0.9,
- Self-similar traffic load steps of $\rho = 0.1$ to 0.5 then from 0.5 to 0.7 with $H = 0.75$,
- Trace packets mean and variance analysed over measurement intervals of 10 seconds and then 100 seconds.

Results are plotted on log-scale since this makes the steps easier to see.

6.7.2 Simulation results for detection of steps

Figures 6-28 and 6-29 show ten 10-second estimations of mean and variance from trace packets, i.e. total time of the plot is 100 seconds, with the step in load occurring at the start of the interval 5, with Poisson traffic. Figures 6-30 and 6-31 show the same estimations for self-similar traffic. Note that as trace packets are sent with a frequency of 0.56 packets / second, the estimations are based on only 56 trace packets per plot.

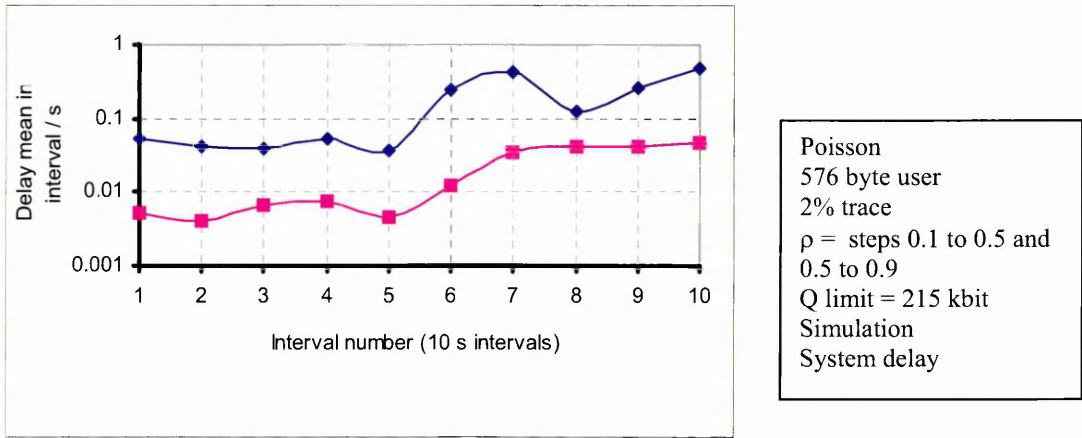


Figure 6-28 10*10-second estimations of mean, Poisson traffic

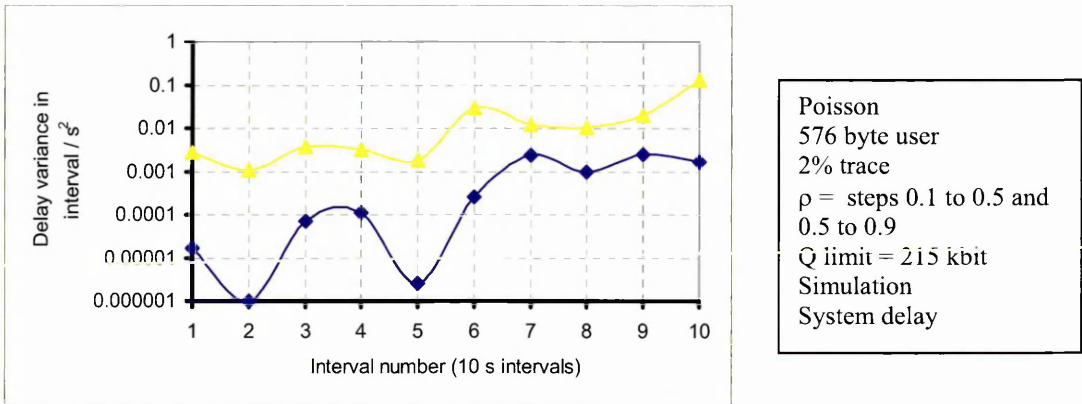


Figure 6-29. 10*10-second estimations of variance, Poisson traffic

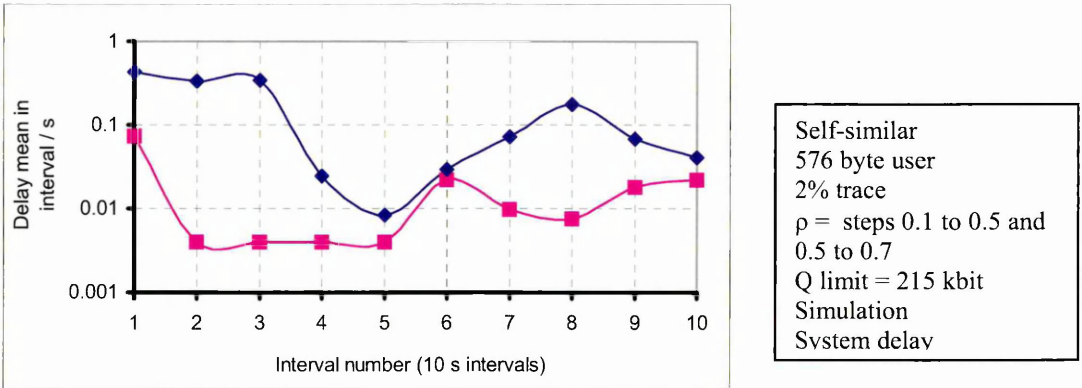


Figure 6-30 10*10-second estimations of mean, self-similar traffic

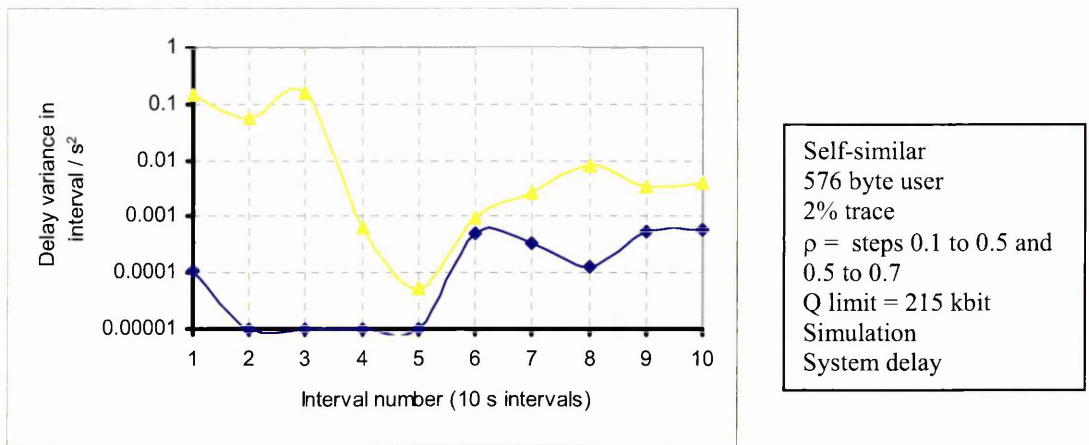


Figure 6-31 10*10-second estimations of variance, self-similar traffic

Whereas the step in delay mean and variance is can be detected by eye in figures 6-28 and 6-29 for the Poisson case, the step is not clear in the self-similar case in figures 6-30 and 6-31. In fact, two of the plots for the self-similar case appear to be going the wrong way. It is postulated that this is because of the few trace packets are being used for each interval and all estimations will be unreliable. If the measurement intervals are increased to 100 seconds, as in figures 6-32 to 6-35, the number of trace packets used in each plot is increased to over 500 and the step becomes even clearer for Poisson and a discernible step emerges for self-similar traffic.

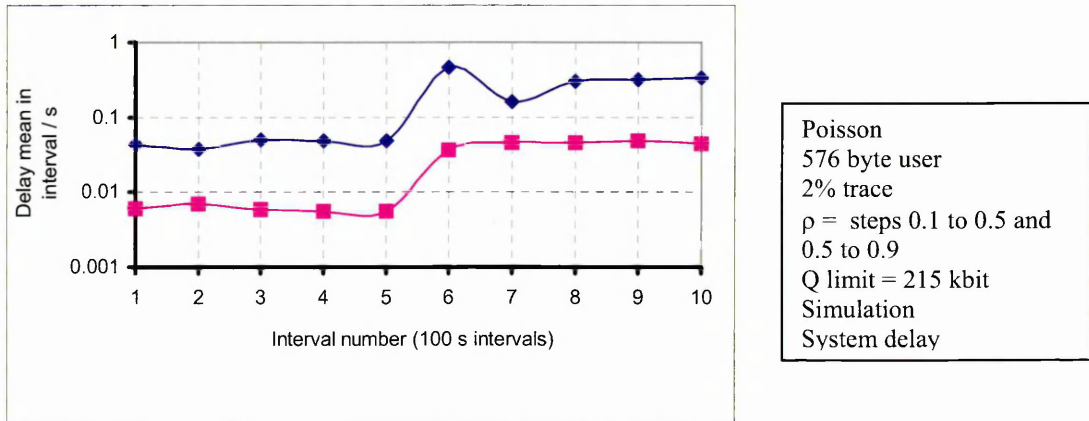


Figure 6-32 10*100-second estimations of mean, Poisson traffic

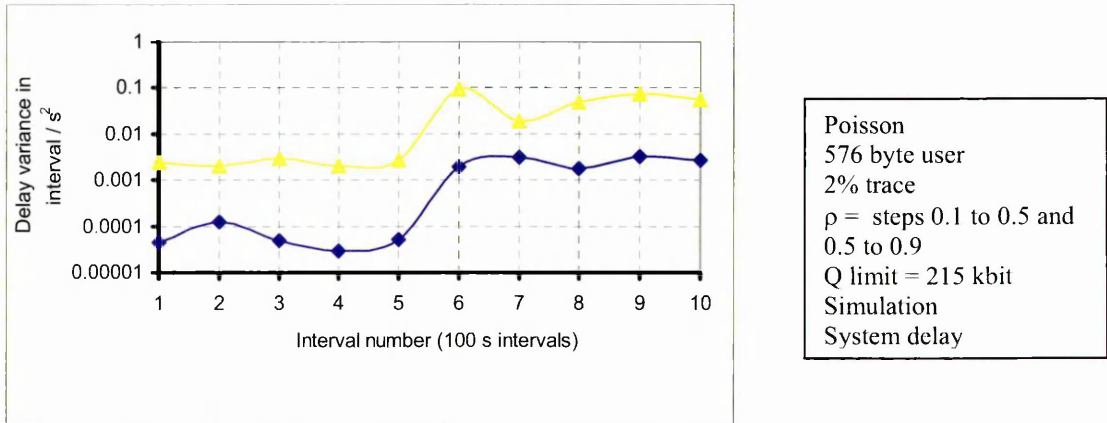


Figure 6-33 10*100-second estimations of variance, Poisson traffic

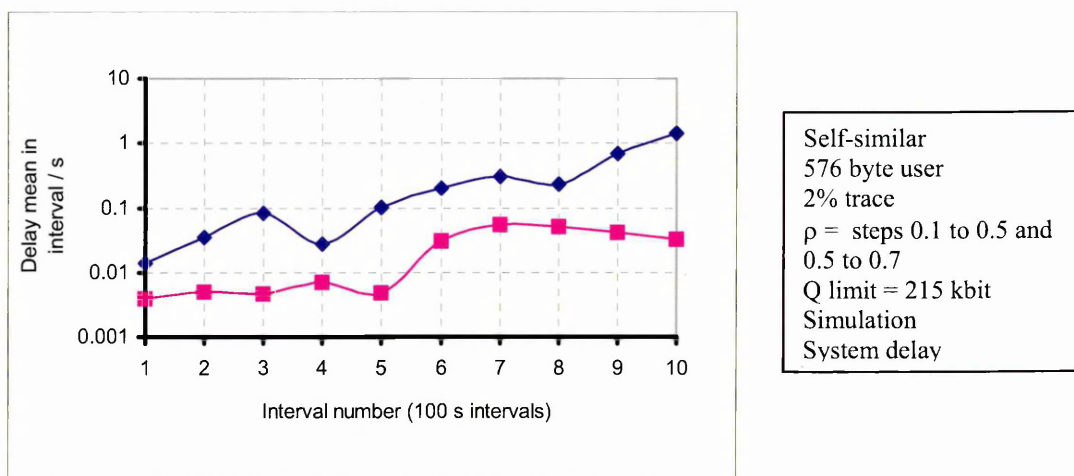


Figure 6-34 10*100-second estimations of mean, self-similar traffic

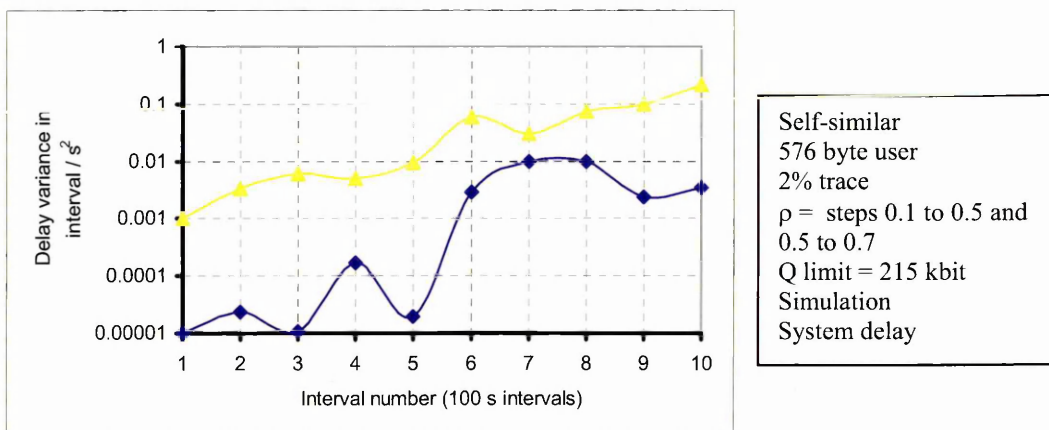


Figure 6-35 10*100-second estimations of variance, self-similar traffic

These results indicate that the use of 10-second estimation intervals resolved the steps in Poisson traffic, but 100 second intervals are required to resolve steps in self-similar traffic. Estimations over four intervals put the detection time at 40 – 400 seconds.

6.7.3 Test network results for detection of steps

Figures 6-36 to 6-41 were taken from the test network, with the same conditions as the simulation, i.e. 2% trace packets, user IP packet length = 576 byte and trace packet length = 64 byte. There are no plots for 10 second intervals for self-similar traffic, as these carried no useful information; just as in the simulation, the steps were not discernible. Table 6-11 shows which plots should be compared.

Measurement periods, traffic type, parameter	Simulation	Test network
10 seconds, Poisson traffic, mean	Figure 6-28	Figure 6-36
10 seconds, Poisson traffic, variance	Figure 6-29	Figure 6-37
10 seconds, self-similar traffic, mean	Figure 6-30	Not plotted
10 seconds, self-similar traffic, variance	Figure 6-31	Not plotted
100 seconds, Poisson traffic, mean	Figure 6-32	Figure 6-38
100 seconds, Poisson traffic, variance	Figure 6-33	Figure 6-39
100 seconds, self-similar traffic, mean	Figure 6-34	Figure 6-40
100 seconds, self-similar traffic, variance	Figure 6-35	Figure 6-41

Table 6-11 Figure comparison table for steps

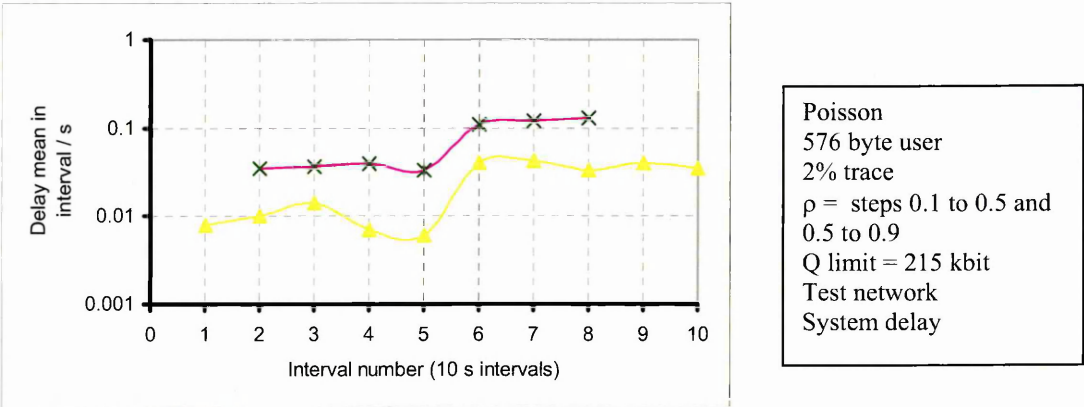


Figure 6-36 10 * 10 second estimations of delay mean, Poisson traffic

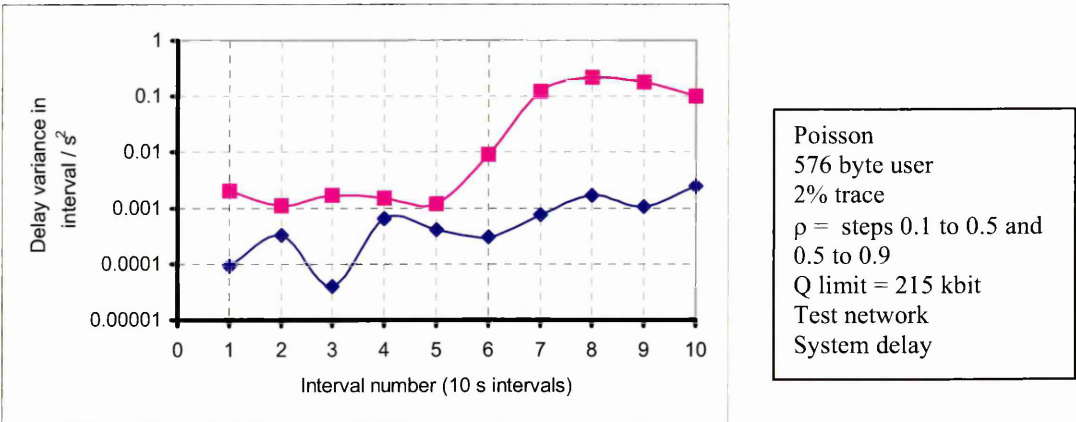


Figure 6-37 10 * 10 second estimations of delay variance, Poisson traffic

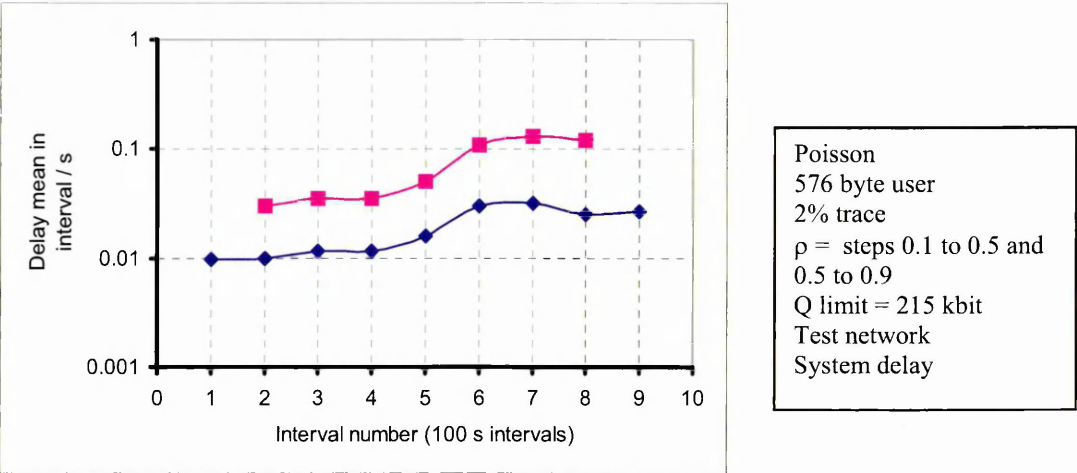


Figure 6-38 10 * 100 second estimations of delay mean, Poisson traffic

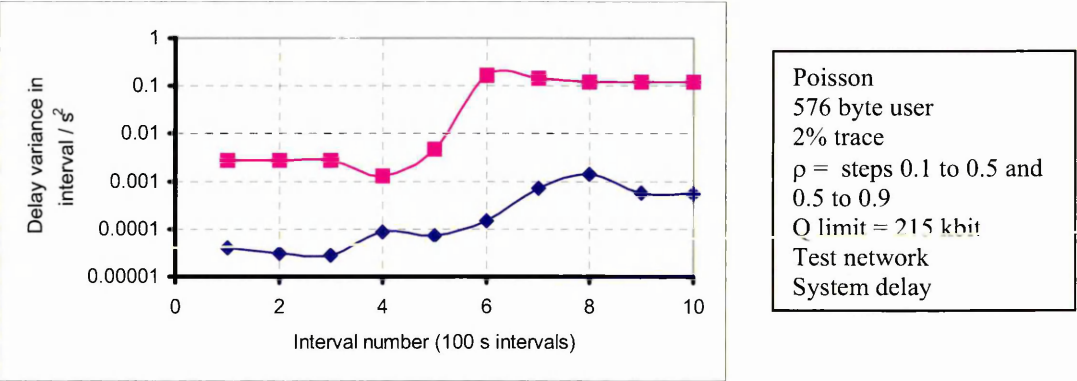


Figure 6-39 10 * 100 second estimations of delay variance, Poisson traffic

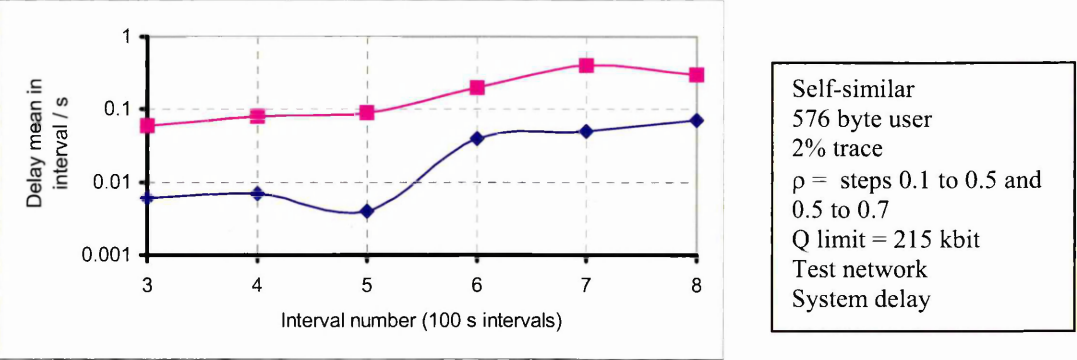


Figure 6-40 5 * 100 second estimations of delay mean, self-similar traffic

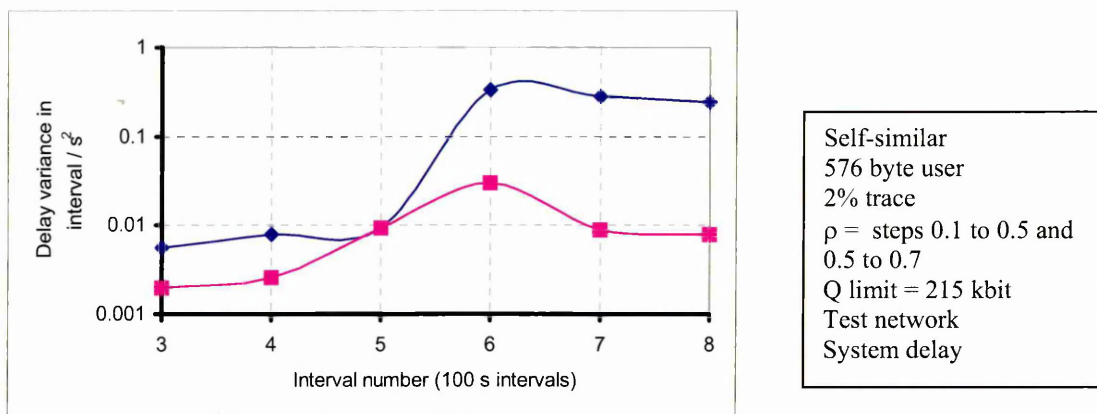


Figure 6-41 5 * 100 second estimations of delay variance, self-similar traffic

The test network results are comparable with the simulation results and confirm that 10 second measurement intervals for the Poisson source and 100 second measurement intervals for the self-similar source are required in order that the steps can be resolved.

6.7.4 Discussion on detection of step changes in load

The conclusions from this section are that, for this particular queue and traffic scenario:

- 10 – 100 seconds is required as a measurement interval, depending on degree of self-similarity in order to resolve step changes in load,
- to actually detect that a step has happened requires at least four intervals (i.e. 40 – 400 seconds). This compares to a queue fill time of around 6 seconds, so that packets will be lost unless the queue length can be increased by a factor of around 7 – 70 for this particular system,
- the measurement times comply with the 1 minute recommended in Y.1541 only when the traffic is Poisson or nearly Poisson. An input to the standards bodies from this study could be to propose a measurement time of 1 – 7 minutes depending on degree of self-similarity of traffic,
- packet loss will be greater for self-similar traffic than for Poisson traffic, firstly because the queue overloads faster and more often and secondly the measurement process takes longer to detect the increase in delay and delay variation.

It is a case of choosing the minimum amount of trace packet intensity for the particular network and traffic patterns encountered. Faster ways of detecting a change in load may include increasing trace packet intensity or even resorting to tagging and monitoring every user packet. Both of these methods result in increased overhead in terms of traffic and processing power at the remote network. The investigation into measuring steps in load is left at this point. Further work is to evaluate sensitivity to packet length and trace packet intensity for various queue and traffic pattern scenarios.

6.8 Use of trace packets compared to other methods: direct sampling and SNMP

6.8.1 Comparison with direct sampling

Results so far in this chapter have shown that trace packets estimate delay and packet loss with an accuracy that is dependent on packet length and queue load.

The use of direct sampling of user packets has not been simulated nor measured in this study, since this method is well documented by other workers (e.g. Zseby [6-11] and

Claffy [6-12] already discussed in chapter 4) where classical sampling theory has been used to calculate confidence intervals.

It is proposed that direct sampling cannot be used to estimate packet loss. The reason is that lost packets have to be counted, which means that every packet would require decoding and it is then no longer a sampling process. Therefore, direct sampling will be considered only for packet delay mean and variance. The inability of direct sampling to detect packet losses renders it not well suited to QoS parameter estimations at high queue loads, which is the condition under which most packets are lost.

The reason for this exercise is to determine where direct sampling may be preferred to trace packets, for instance at lower queue loads, where greater accuracy may be achieved with tolerable management traffic overhead and remote network processor power. The aim is to give guidelines on which estimation method should be mapped onto which management mode, i.e. normal, intermediate, damage limitation - and where the thresholds for changing the method should be in terms of queue load ρ .

The use of direct sampling carries the following assumptions:

- all user packets are tagged on entry to the network, containing the time of tagging,
- sampling is done as packets pass by, so that it is without replacement,
- the mean delay estimation is normally distributed about the true mean,
- the delay variance estimation varies about the true variance according to the chi-square distribution.

These last two assumptions hold even though the population distribution is not normal by the action of the central limit theorem, provided that the sample size n is large enough and the samples are independent. Kreysig [6-7] recommends $n > 30$.

Confidence in the estimation of delay mean

Let the estimated delay mean be ϕ and the true value be ψ . There will be a range of values around ϕ that has a probability P of containing the true value ψ . The wider the range, the higher will be the probability that ψ lies within it. This probability can be written as

$$P(\phi - c_1 \leq \psi \leq \phi + c_2) = \gamma \% \quad (6-1)$$

where c_1 and c_2 are the lower and upper limits of the range. The confidence interval is the distance between c_1 and c_2 and the confidence level is γ . A suitable value for γ is taken to be 95% for this study.

The error probability distribution is assumed to be normal, which is a symmetrical distribution so that $c_1 = c_2$ and the interval is $2c$.

The normal distribution is generated by

$$f(x) = \frac{1}{\xi\sqrt{2\pi}} e^{-\frac{(x-\bar{x})^2}{2\xi^2}}$$

When used in calculating confidence intervals with an unbiased estimator, the mean error \bar{x} is set to zero and the standard deviation ξ is set to 1 and this case is known as the standard normal distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (6-2)$$

Let $d, -d$ be the x -axis values corresponding to γ % of the area of the curve described by (6-2). Values of d are published in tables [6-7]. For $\gamma = 95\%$, $d = 1.96$.

The value d is used to calculate the confidence interval using the following formula [6-8].

$$c_1 = c_2 = \frac{d\sigma}{\sqrt{n}} \quad (6-3)$$

where
 n = sample size
 σ = population standard deviation

This formula can only be used when the population standard deviation σ is known. In most cases σ is not known and has either (a) to be estimated as a separate process or (b) the sample standard deviation can be used instead with the Student-t distribution. In the present study the user packet delay variance σ^2 is known since it has been simulated and measured so that σ can easily be determined.

Table 6-12 contains the 95% intervals, calculated by using the absolute values of mean and variance from figure 6-11 and 6-12 respectively. The sample size has been assumed to be equal to the number of trace packets at an intensity of 1% of the user mean.

Load ρ	User packet mean from figure 6-11 / s	User packet std deviation from figure 6-12 / s	Confidence interval / s	Number of samples = number of trace packets	Confidence interval / % of user mean	Trace estimation accuracy / %
0.1	0.002	0.00737	0.0024	278	+/- 36	- 2
0.3	0.0077	0.0156	0.0051	833	+/- 12	- 9.8
0.5	0.0179	0.0272	0.0088	1389	+/- 7	- 3.7
0.7	0.041	0.0516	0.017	3889	+/- 3.3	- 2.7
0.9	0.16	0.171	0.056	5000	+/- 2.8	- 0.22
0.95	0.322	0.306	0.101	5280	+/- 2.3	-0.36

Table 6-12 Comparison of estimation accuracy of delay mean with confidence intervals

Table 6-12 shows that estimation accuracy by trace packets is well inside the 95% confidence intervals for direct sampling, whatever the load.

Confidence in the estimation of variance

For variance, the chi-square distribution (χ^2) is usually used [6-7]. This is not a symmetrical distribution, so the upper and lower confidence interval limits are not the same distance from the mean and have to be calculated separately. Figure 6-42 shows a typical shape of a chi-square distribution. The area under the curve between the confidence interval limits is $(1-\alpha)$, so that α represents the complement of the confidence level. Equal areas lie to the left of the lower limit and to the right of the upper limit; the total area is 1.

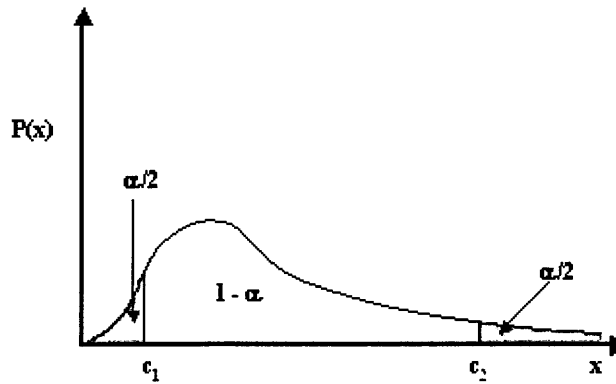


Figure 6-42 Chi-square distribution

The chi-square distribution shape depends upon the degrees of freedom ($= n-1$). As n increases, the distribution becomes more of a normal shape.

To obtain a confidence interval, it is necessary to find the upper and lower percentiles of the distribution so that [6-8]

$$P\left[\chi^2_{n-1, \alpha/2} \leq \frac{(n-1)s^2}{\sigma^2} \leq \chi^2_{n-1, 1-\alpha/2}\right] = 1-\alpha$$

and re-arranging

$$P\left[\frac{(n-1)s^2}{\chi^2_{n-1, 1-\frac{\alpha}{2}}} \leq \sigma^2 \leq \frac{(n-1)s^2}{\chi^2_{n-1, \frac{\alpha}{2}}}\right] = 1-\alpha \quad (6-4)$$

where P = probability, n = sample size, $(1-\alpha)$ = confidence level, σ^2 = population variance and s^2 = sample variance.

If a confidence level $(1-\alpha) = 0.95$ is again assumed, then $\alpha = 0.05$ so that $\alpha/2 = 0.025$ and $(1 - \alpha/2) = 0.975$.

Again a sample size n equal to the number of trace packets at 1% intensity is assumed. Values of $\chi^2_{n-1, 0.025}$ and $\chi^2_{n-1, 0.975}$ were looked up in standard tables and converted into the lower and upper confidence interval limits by dividing into $(n-1)$ as specified by equation (6-4) and shown in table 6-13.

Load ρ	User packet variance from figure 6-12 / s^2	Number of samples = number of trace packets	Confidence interval / % of user variance	Trace estimation accuracy / %
0.1	0.000054	278	+19, -15	- 7.5
0.3	0.000246	833	+10, -9	- 18.3
0.5	0.00074	1389	+8, -7	- 8.3
0.7	0.00266	3889	+4.6, -4.3	- 5.4
0.9	0.029	5000	+4.0, -3.8	- 2.9
0.95	0.11	5280	+3.9, -3.7	- 0.7

Table 6-13 Comparison of estimation accuracy of delay variance with confidence intervals

Tables 6-12 and 6-13 show that estimation accuracy achieved with trace packets fall further inside direct sampling confidence intervals as the load increases. The exceptions to this for both the mean and variance are at $\rho = 0.1$ where the trace packet method appears to provide highly accurate estimations. However this is considered to be unreliable due to the oscillatory nature of the estimation for such low numbers of trace packets, as illustrated in figures 6-11 and 6-12.

If these estimation accuracies are compared with results from Drobisz et al in table 4-4, they achieved 10% accuracy in delay mean and variance with a sampling accuracy of 2%. They do not specify the network load for this accuracy, but they did require stratification to achieve it. This experiment has shown that trace packets can give a greater than 10% accuracy at 1% intensity for $\rho > 0.7$.

6.8.2 Comparison with SNMP commands

It is interesting to briefly compare the overhead of trace packets with SNMP transactions. The in-band management overhead traffic that flows with 2% trace packets at a link rate of 128kbit/s and 576 byte user packets depends upon the load but is a maximum of 0.5 trace packets / second, so that the channel is sampled with a period of just under 2 seconds. This equates to approximately 36 bytes / second.

According to section 4.4.1, SNMP 'Get' commands use 350 bytes in each direction to retrieve a single parameter value. If these were used at an equivalent rate of every 2 seconds, for example to retrieve router queue fill, the overhead would be 175 bytes in each direction per second, which is about 5 times the overhead associated with trace packets.

6.9 Summary of the chapter

This chapter has illuminated several key points and this final section gathers them together and offers some discussion.

6.9.1 Measurement modes

Results from previous sections show that the most likely traffic patterns to be encountered on practical networks are self-similar and that trace packets show an advantage over direct sampling at ρ of around 0.5 – 0.7, at least when the user packet length is 576 bytes. Therefore these values of load can be used as a threshold for changing the measurement mode. Table 6-14 is a development of table 4-1, containing more information from the experiment on the relationship between management mode and network load.

Management mode	Criteria	Actions	Appropriate estimator
Normal	Low network load, $0.1 \leq \rho < 0.5$	None	Trace packets can be used, but direct sampling gives greater accuracy
Intermediate	Medium network load, $0.5 \leq \rho < 0.7$	Increase bandwidth, find alternative routing	Trace packets and direct sampling give approximately equal accuracy
Damage limitation	Network load too high $\rho \geq 0.7$	Urgent measures to limit SLA damage	Trace packets give highest accuracy

Table 6-14 Management mode guidelines

In ‘normal’ mode, direct packet sampling is appropriate, with accuracy determined by sampling theory as discussed in the previous section.

In ‘intermediate’ mode, either direct sampling or trace packets can be used as estimators and which is best is determined by factors such as the degree of traffic self-similarity and the processing power available at the remote network.

In ‘damage limitation’ mode, use of trace packets is appropriate, since this has been shown to give increasing accuracy with higher loads and will provide a more direct indication of packet loss. This is survival mode, where configuration commands and other essential management traffic must be given high priority in queues. This high priority event traffic must itself be kept small to minimise further impact on packet losses.

For lower loads, more accuracy can be gained by using direct sampling but at higher overhead and processing cost. Further refinement of these thresholds, their confirmation at different user packet lengths and desirable degrees of hysteresis in measurement and management actions is left for further work.

6.9.2 Comparison of behaviour of self-similar and Poisson traffic

Self-similar traffic is badly behaved compared to Poisson traffic, having long-tailed delay pdfs when queued, in line with its more slowly-decaying variance with time-scale. When the queue is 100 Mbit, the delay pdf extends up to very large and unmanageable values. The pdfs of delay exhibit double peaks when a queue limit is imposed. It is not practical to load networks with real routers to more than $\rho = 0.7$ with self-similar traffic (at least at $H \geq 0.75$) as the delay variance and packet loss becomes excessive. Limiting the queue length reduces the delay mean and variance at the expense of some packet loss (the packet loss is greater with self-similar traffic).

6.9.3 The use of trace packets as estimators of packet delay and loss

A trace packet method of estimating delay mean and variance has been evaluated and a view given on the relationship between estimation accuracy and trace packet intensity for Poisson and self-similar traffic for popular user packet lengths. The evaluation has compared simulation results with measurements on a test network with good agreement. The simulation is an ideal abstraction, whereas the test network exercise was valuable in that it revealed issues such as timing constraints and processing jitter.

Estimations of packet delay are most accurate for shorter packets and high queue loads. Packet loss can be detected directly using trace packets, but the losses are under-estimated

and the degree of under-estimation is worse for self-similar traffic. The development of a rigorous theory for this interesting behaviour is not attempted, but it seems as though trace packets give higher accuracy estimations when more of them become trapped behind user packets in the queue. This may lead to more of them being subjected to the turbulence in the queue and hence making a more accurate estimation of the turbulence, which will increase with load and with user packet length. Estimates of delay made using direct sampling, on the other hand, will become less accurate as the turbulence increases, due to the confidence interval length being proportional to the standard deviation (for a fixed sample size). Further work is required to extend queue theory to include trace packets in both Poisson and self-similar cases and to develop a rigorous theory to describe the relationship between accuracy and degree of self-similarity.

Knowledge of the pdf shape, which can be measured over a period time using trace packets to form a histogram, along with the measure of mean and variance, can be used to determine and report on the delay variation intervals such as percentiles or quantiles, as recommended in IETF and ITU-T standards on QoS parameters, on a path-by-path basis.

Comparisons with direct sampling indicate that trace packets should be used as estimators for queue loads of $\rho > 0.5$, with direct sampling used below this. Evaluation of the thresholds for other packet lengths and required hysteresis levels are suggested as further work. Trace packet estimation uses about one-fifth of the overhead that would be consumed by SNMP commands to interrogate each router queue

Detection of steps in load can be made using trace packets, but the detection time with parameters chosen for these tests is around 20 times too long to avoid packet loss if the step is sudden. Compliance with the 1 minute measurement time recommended in Y.1541 can only be achieved with near-Poisson traffic.

6.10 References for chapter 6

- [6-1] K Claffy. *The nature of the beast, recent Traffic Measurements from an Internet Backbone*. CAIDA, April 1998.
- [6-2] Walter Willinger, Murad Taqqu, Robert Sherman and Daniel Wilson. *Self-similarity through high variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level*. IEEE Transactions on Networking, Vol 5, No1, February 1997. Pages 71 – 86.
- [6-3] M Jolly and K Briggs. *Analysis and Simulation of Internet Round-trip Times*. BT Exact report to Eurescom, February 2003. Available from www.btexact.com/projects/complexity
- [6-4] CJ Bovy, H Mertodimedjo, G Hooghiemstra, H Uijterwaal, P Van Mieghem. *Analysis of End-to-end Delay Measurements in Internet*. Delft University of Technology, 2001.
- [6-5] A Bragg. *Generating Self-Similar traffic for OPNET Simulations*. Fujitsu Network Communications. Undated but later than 1997.
- [6-6] B Zhou and Z Sun. *Simulation study of self-similar traffic and network performance*. Centre for Communications Systems Research, University of Surrey. Undated, but known to be after 2000.
- [6-7] E Kreysig. *Advanced Engineering Mathematics*. John Wiley and Sons, 1979. Page 904 and table A12.
- [6-8] Dang Quang and Bui The Hong. Institute of Information Technology, Hanoi. See <http://www.netnam.vn/unescocourse/statistics/statistics.htm>. Link checked 5th December 2004.
- [6-9] I Cozzani and S Giordano. *Traffic sampling methods for end-to-end QoS evaluation in large heterogeneous networks*. Computer Networks and ISDN Systems 30 (1998), pages 1697 – 1706.
- [6-10] ITU-T. *Network Performance Objectives for IP-based Services*. ITU-T Recommendation Y.1541, May 2002, page 7.
- [6-11] T Zseby. *Deployment of Sampling Methods for SLA Validation with Non-Intrusive Measurements*. Fraunhofer Institute for Open Communications Systems (FOKUS). Undated.
- [6-12] K Claffy and G Polyzos. *Application of sampling methodologies to Network Traffic Characterisation*. Computer systems laboratory, University of California. Not dated, but known to be after 1993.

7 Discussion and conclusions

This chapter draws together the conclusions of this study and discusses the implications of the findings.

7.1 The changing role of satellite links and implications on OSS

Satellites are naturally suited to broadcast and multi-cast services because their high vantage point affords them a large coverage area. Users can be added quickly without changes to the infrastructure, which makes them suited to fast introduction of services.

The advantages with satellite links are tempered by high latency (about 310ms per hop for a GEO satellite including modem delays), high cost, a lifetime of typically 10 – 15 years and limited power and bandwidth.

Most satellites are ‘transparent’ in that they do not demodulate the signals and most are in GEO. Processing satellites and other orbits have been tried with limited commercial success, mainly because of the larger investment required before service can start. Access to satellite capacity from earth stations is managed using time and / or frequency slots or different codes. Whatever the scheme, a scheduler is necessary which will add delay variation due to data queueing at the earth stations awaiting transmission.

Satellites have always been used for television broadcast and for maritime and aeronautical communications. They are being increasingly used for IP data to the user, for both broadband access to the Internet and for connection to private networks.

Currently, OSS used on satellite networks is fragmented and mostly at the element management level. OSS on terrestrial networks has reached a higher degree of integration on fault and alarm correlation, work-flow and provisioning and customer ordering / fault reporting. There is a drive for greater OSS integration coming from the wider telecommunications context. Network operators are currently developing next generation (or ‘converged’) networks (NGN) with a high bandwidth core network that carries data from any service and a range of access networks that will enable users to connect anywhere using a wide variety of terminal types. The services offered will include voice, video / audio streaming, broadband connections to the Internet and Intranets and web-based services. The OSS that manage these networks and services will have a distributed architecture and will themselves use the web for communicating between clients and servers.

Satellite links fit into NGN in three main roles; first in inter-core network connections, secondly in intra-core for distributing data to the network edges and thirdly in the access network. The first of these is declining, the second is small and static and the third is growing; the challenge for satellite communications is that the growth in the access network is sufficient to offset the decline in inter-core connect. Therefore the role of satellite links is migrating more towards the user or access networks, especially where it is not economical or not possible to deploy fibre or copper. This migration means that satellite is moving from few high-revenue users on large networks, towards many low-revenue users connected to small networks.

Changes have also been occurring with satellite operators business models; de-regulation and migration of the market-place has motivated satellite co-operatives to privatise and stimulated several spin-off operators. This has resulted in increased competition and a greater choice for service providers.

Considering the above, an OSS for satellite links must be able to sign up customers quickly and reliably, be able to support broadcast and multi-cast services to small networks with limited processing power and keep the overhead traffic as low as possible. The increasing number of satellite operators, service providers, network operators developing NGN, and diverse users are demanding integration and automation of OSS. Such integration and automation at lowest capex and operational costs can only be achieved if OSS components conform to open standards and industry forum models.

7.2 Evaluation of differences between OSS for terrestrial and satellite networks and services

The provision of managed networks and services demands the use of Service Level Agreements between actors in the value chain, which must contain the grades of service offered and define management boundaries, especially at user premises. Service providers want to provide managed end-to-end services that sometimes requires reaching through other actors in the value chain. Examples are managed routers or satellite receivers at user premises for bundled services or TV broadcast reception.

Increasingly the integration and automation of OSS has been progressed through standards bodies and industry forums as expected. An evaluation of these reveals that the TMF eTOM is the most advanced and supported business process framework. The TMF have also published guidelines on NGOSS (Next Generation OSS) suited to NGN, using a distributed hub-based architecture that scales well and minimises the cost of developing software and interfaces. An evaluation of the eTOM processes has been carried out for satellite links, first by identifying which processes are different for satellite and terrestrial networks and then ranking the differences.

This evaluation revealed that the most significant differences between satellite and terrestrial network OSS are at the physical layer as this is where most satellites operate. In particular, the management and measurement of QoS parameters ranked the highest in terms of impact on the management overhead traffic.

7.3 Measurement and management of QoS parameters

In order to determine how an OSS can best measure QoS parameters over a path including a satellite link with minimum overhead and processing power, a brief evaluation has been carried out into measurement methods and tools. It is concluded that active measurements are more suited to satellite because they require much less processing at the remote end, with the penalty of adding a small amount of in-band overhead traffic.

QoS parameters relevant to satellite communications are defined in ITU-T and IETF standards; these include throughput, packet delay, delay variation, loss, mis-insertion and error rate. The actual QoS parameters chosen for measurement in this study are those related to layer 1 (physical) performance associated with satellite links, i.e. delay, delay variation and packet loss. In the standards, these QoS parameters are given a range of limiting values according to specified grades of service. The delay limit is given as a mean, the delay variation is given as a percentile or quantile of the delay pdf and the loss is given

as a fraction. Of these, the most tricky is the delay variation, since the delay pdf shapes are typically not normal, especially if the packet arrival rates at the satellite link have a degree of self-similarity.

The notion of trace packets is introduced to measure delay, delay variation (including an estimation of the pdf shape) and packet loss on a network path that includes a satellite link. The idea of trace packets in an IP network is similar to the use of OAM cells in an ATM network, where they are used in a number of defined OAM ‘flows’, the top flow of which is end-to-end (F5) accompanying user cells in the same virtual circuit.

7.4 Simulation model and test network

In investigating a suitable queuing theory, it is discovered that the M/M/1 system is convenient mathematically but does not effectively model a packet system, since it implies that some packets are zero length. M/D/1 is suited much better to packet transfer over satellite, but the delay pdf for this queue cannot be obtained through the usual use of the Laplace transform. Therefore, M/D/1 is used for the experiments but the queue is made temporarily M/M/1 for calibration purposes. In the course of simulations using the OPNET modelling tool, a view is given on the delay pdf of M/D/1 as a novel result. The M/D/1 is suited to Poisson packet arrivals which has been shown not to model IP traffic well; Internet traffic on real networks has a degree of self-similarity with a Hurst parameter of around 0.75. For self-similar packet arrivals the G/D/1 queue is a better model but this queue is mathematically difficult and the theory is not pursued. Instead, experiments are performed using Poisson and self-similar traffic using simulation and a test network.

The construction and testing of a real network was a very worthwhile part of the work, discovering the extent to which real networks can introduce features that are not included in the simulation. These features included timing jitter, changing the order of packets, generation of unwanted packets (mainly ARP requests) and synchronisation problems that made it difficult to detect when packets were dropped. To mitigate these, the jitter was analysed, the queuing discipline in the routers was forced to FIFO and unwanted packets were filtered out. In addition, the simulation is capable of running many thousands of seconds of measurement time, whereas the test network is limited by the number of packets that can be captured (64,000), so that the test network results did not always have the capacity to converge. The test network revealed that routers have queues with a length limit of around 215 kbit. This figure was adopted into the OPNET simulations. In order to perform the experiments, it was necessary to construct and test Poisson and self-similar user packets sources and trace packet sources for both the simulation tool and the test network. In the case of the test network, this was done using a PC running Linux with a software program called RUDE freely downloaded from Sourceforge.

A single queue and server system is used in both the simulation and the test network, to emulate an entire path. The justification for this is that the queue just before the satellite link will be the dominant queue in the path, assuming that the satellite link is a constriction. In general this will be the case, since the satellite capacity is the most expensive part of the path.

7.5 The use of trace packets as estimators

Experiments have been conducted using the simulation tool and on the test network in order to verify that trace packets can be used to measure the QoS parameters of delay,

delay variation and loss suffered by IP packets and to identify some bounds over which such measurements are valid.

It was discovered that trace packets can estimate delay and loss to user packets, with an accuracy that depends upon the network load and on the traffic type. The most important result is that the accuracy of estimation provided by trace packets does not reduce with increasing queue load, which is a good result for satellite links with expensive bandwidth and limited processing at the remote end. The accuracy is better for Poisson than for self-similar traffic due to the larger variance of the latter, especially in the estimate of packet loss. Example accuracies are that estimates within 10% of the true values of delay mean and variance are achieved for loads of 0.7 (self-similar) or 0.9 (Poisson) for 2% trace packet intensity. User packet sizes are taken from published Internet results of the three most common sizes; these are around 576 bytes, 1500 bytes and 64 bytes, with the majority of the results taken with 576 bytes since this size carries the most byte volume. The length of the trace packets is fixed at 64 bytes to give space for a time-stamp and sequence number in the payload. The generation pattern of the trace packets is constant.

With the use of trace packets, distributed servers on small remote networks can collect delay and loss information and report back either by satellite or by terrestrial means during quiet periods or when interrogated. Alternatively a server can raise an event report (such as an alarm) which is sent immediately. The use of trace packets should scale well with multi-cast and broadcast services to which satellite is naturally suited – and should map well onto emerging NGOSS architectures. Monitoring can be performed on the small remote networks, which can report periodically using dial-up or satellite return links. Not all users need to have a return channel; the management system can make a decision on measurement method from a small sample of users. If there is no return channel from a particular remote network, an indication of data losses and delay profile can still be given to the user.

The trace packet method saves on processing power at the remote end when compared to direct sampling. Since trace packets can carry time-stamps and sequence numbers, the delay pdf can be estimated from them and the mean and variance measured. Then, from this knowledge the quantiles or percentiles can be calculated for reporting the delay variation into the management system in the form recommended by the standards.

The trace packet method of estimation is compared with alternatives, i.e. obtaining estimates using direct user packet sampling and with using SNMP commands to interrogate the router queue. From other experiments, it is calculated that using SNMP commands is much more expensive on management overhead traffic, i.e. about 5 times that required for trace packets. The accuracy that would be obtained with direct sampling has been calculated using confidence intervals, since classic sampling theory will apply. Direct sampling estimations of delay will reduce in accuracy with increasing load (for the same sample size) since the variance is increasing, whereas trace packet estimation does not reduce in accuracy with increasing load (for the same number of trace packets). Also, as the load increases the processing power required with direct sampling also increases. A criticism of the trace packet method of estimation is that it requires packet order to be maintained along the path; this could be mitigated by use of FIFO queues or by employing trace packets on every packet flow.

7.6 Management modes

A pragmatic approach to selection of different measurement techniques in different scenarios and management modes, characterised by queue load, is possible.

OSS management modes have been proposed as 'Normal', 'Intermediate' and 'Damage Limitation', depending upon the queue load as revealed by trace packet variance. Network loads have been assigned to these modes for Poisson and self-similar traffic as a result of the experiments, along with recommendations on measurement and estimation methods. For low network loads, say below 0.5, the use of direct sampling is recommended as this yields higher accuracy without a heavy processing burden. For loads above 0.5, the use of trace packets is recommended as the accuracy of direct sampling reduces and the required processing power reduces at the remote network to detect lost packets. The optimum threshold depends on the level of self-similarity in the traffic flow and further analysis of this, suitable hysteresis and network signalling is left for further work.

7.7 Faster measurement times

The ITU-T recommendation Y.1541 recommends an SLA reporting update interval of 1 minute. TCP sessions time-out after a 35 second break in the transmission over a GEO satellite. The simulations and test network experiments were run over several thousand seconds. With these faster times in mind, additional experiments were run to estimate the shortest possible times to detect a step change in load.

The results of these experiments are that the required measurement time to detect the step is of the order of 40 – 400 seconds depending upon traffic type, with a satellite link rate of 128 kbit/s. Steps in load for self-similar traffic take longer to detect because of the higher variance. This means that unless very large queues are available to buffer overload, some packet loss will occur. In this study, the queue length of 215 kbit filled in less than 10 seconds when a step load of $\rho = 0.3$ to 0.7 of self-similar traffic was applied.

The changes in steps took about 40 – 400 seconds to detect in the experiment. However, if the link rate is increased then the measurement time is reduced; it is not unreasonable to assume that the satellite delay (say 0.3 s) equals the measurement time at a rate of about $(40 \text{ to } 400)/0.3 * 128 \text{ kbit/s} = 17 \text{ to } 170 \text{ Mbit/s}$. These are not unusual rates for commercial satellite links; there are many links of 45 Mbit/s and a few at 155 Mbit/s, which actually represents the state of the art in transmission rate over satellite. These kinds of rates are used for services such as Internet back-bone links and inter-core network connections.

The required queue length in order to allow measurement and management action is at least the sum of the latency plus the detection time, which is say 0.6 s at 155 Mbit/s or 93 Mbit. This is significantly above the 215 kbit measured on a commercial router. The issue of queue length should be highlighted to designers of satellite links perhaps through the FSN industry forum. However the queue length issue does not affect the argument that is advantageous to use trace packets as estimators of packet delay and loss under conditions of high load.

The reporting update interval of 1 minute is only possible with Poisson or near-Poisson traffic. A contribution from this study to the standards bodies on the findings relating to self-similar traffic and satellite links could possibly be made.

7.8 Further work

There are many areas of further research which have emerged from this study. These include:

- Evaluation of the trace packet estimation method with mixed packet sizes. The results in this study have concentrated mostly on one popular size (576 byte) typical of TCP traffic. Different packet sizes are produced by different types of services, e.g. VoIP may typically use shorter packets while streaming services may use longer packets.
- Analysis of use of trace packets with different queue disciplines and multiplexed flows. Queue disciplines like weighted-fair operate on a per-flow basis while core networks tend to multiplex flows together from different services (and hence different packet sizes). The effectiveness of trace packets when injected per-flow should be compared to their effectiveness when injected per multiplex.
- Integration with layer 3 QoS control plane mechanisms using priority queueing such as diffserv. Diffserv allows management policies on packet classification and queueing algorithms to be implemented in routers according to availability of network resources. The integration of these management mechanisms with the trace packet method of measurement could be explored.
- Development of a rigorous queueing theory for a mix of trace and user packets in a G/D/1 queueing system to evaluate accuracy bounds. This requires research into the mathematics of queueing theory to determine queue behaviour with Poisson and self-similar packets when mixed with trace packets.

7.9 Concluding remarks

This thesis has evaluated some of the main differences between OSS requirements from satellite and terrestrial communications networks and shown that these are in the area of QoS management. It has presented results on a trace packet method of estimating user packet delay and loss that is suited to satellite links carrying IP traffic and shown that this has advantages over other measurement methods at high traffic loads. It is hoped that the satellite communications industry will benefit in some small measure from this work.

End of report.

Appendix A OPNET models and code

A.1 Overall model

The top-level OPNET model used for most of the simulations is shown in figure A1

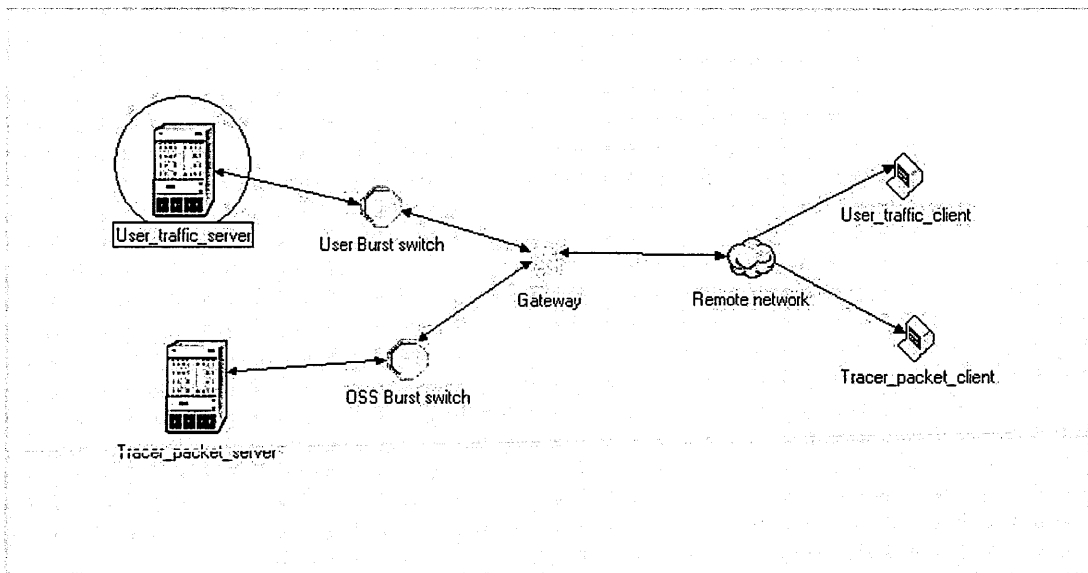


Figure A1 Overall model, network level

OPNET provides the ability to create packets to any formats using a packet editor. For this study, packet formats called `mrf_packet` and `mrf_oss_pkt` were designed. The `mrf_packet` consists of a 20-byte header and variable length payload, while the `mrf_oss_pkt` consists of a 20-byte header and 44-byte payload. The traffic sources were instructed to produce packets to format `mrf_packet` and `mrf_oss_pkt` as appropriate.

The links that come with the OPNET library generally have a transmission speed associated with them that enables modelling of latency when run across geographical locations. In fact the Project wizard in OPNET assumes that a network is to be built across cities and countries and the basis of the design template is a world map. This is not appropriate for this study so a special and very simple link type was created using the OPNET link editor called `mrf_link`, designed to have zero delay, infinite bandwidth and be capable of passing packets of format `mrf_packet` and `mrf_oss_pkt`. Every link in figure A1 is of type `mrf_link`.

A.2 Traffic servers

The traffic servers on the left of figure A1 can produce packets of any length to a range of distributions. The OPNET Version 9 library contains a range of distributions like constant, exponential, Pareto etc that can be loaded at simulation run time. When Poisson traffic is required, the `User_traffic_server` contains a 'simple' packet source that loads an exponential

distribution for the inter-arrival times with a mean of $1/\lambda$. When self-similar traffic is required, the User_traffic_server contains 40 on-off sources with 'on' and 'off' times derived from the Pareto distribution. The design of this compound source is given later in this appendix. The Trace_packet_server is another 'simple' packet source, almost identical to the Poisson version of the User_traffic_server, but loads a constant distribution in order to produce packets evenly spaced in time. The Poisson and Trace packet sources are described together since they are based on the same OPNET process model.

A.2.1 Poisson and trace packet source

The process model for these sources is based on an OPNET library process called 'simple packet generator', but this has been modified to produce a 'step' in the rate at which packets are produced in order to simulate step changes in load described in chapter 6. The step time and the rates before and after the step are set as simulation parameters. If no step is required then the step can be set to occur after the simulation has finished. The process model is shown in figure A2 and the fragments of C code, called executives in OPNET, are given where changes have been made to the library process model 'simple packet generator'.

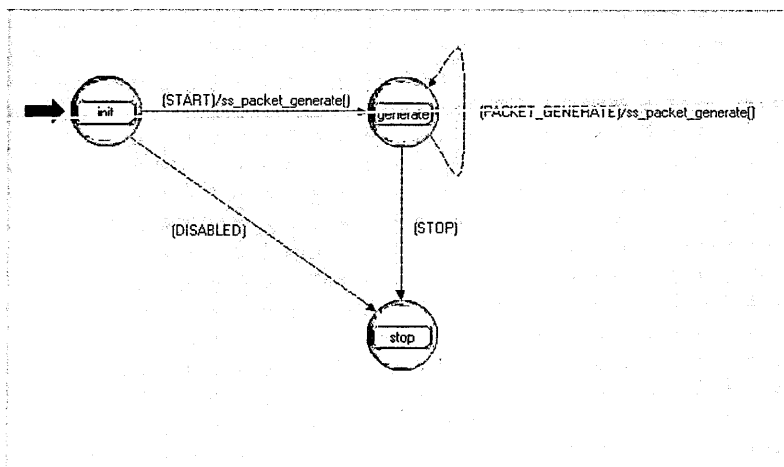


Figure A2. Packet generator process model

The Init process enter execs are given below:

```

/* At this initial state, we read the values of source attributes */
/* and schedule a selt interrupt that will indicate our start time */
/* for packet generation.
*/

/* Obtain the object id of the surrounding module.
*/
own_id = op_id_self ();

/* Read the values of the packet generation parameters, i.e. the
/* attribute values of the surrounding module.
*/
op_ima_obj_attr_get (own_id, "Packet Interarrival Time", interarrival_str);
op_ima_obj_attr_get (own_id, "Packet Interarrival Time 1", interarrival_str_1);
op_ima_obj_attr_get (own_id, "Packet Size", size_str);
op_ima_obj_attr_get (own_id, "Packet Format", format_str);
op_ima_obj_attr_get (own_id, "Start Time", &start_time);
op_ima_obj_attr_get (own_id, "Stop Time", &stop_time);
op_ima_obj_attr_get (own_id, "Step Time", &step_time);

```

```

/* Load the PDFs that will be used in computing the packet */
/* interarrival times and packet sizes.
*/
interarrival_dist_ptr = oms_dist_load_from_string (interarrival_str);
interarrival_dist_ptr_1 = oms_dist_load_from_string (interarrival_str_1);
pksize_dist_ptr = oms_dist_load_from_string (size_str);

/* Verify the existence of the packet format to be used for */
/* generated packets.
*/
if (strcmp (format_str, "NONE") == 0)
{
/* We will generate unformatted packets. Set the flag. */
generate_unformatted = OPC_TRUE;
}
else
{
/* We will generate formatted packets. Turn off the flag. */
generate_unformatted = OPC_FALSE;

/* Get the list of all available packet formats. */
pk_format_names_lptr = prg_tfile_name_list_get (PrgC_Tfile_Type_Packet_Format);

/* Search the list for the requested packet format. */
format_found = OPC_FALSE;
for (i = prg_list_size (pk_format_names_lptr); ((format_found == OPC_FALSE) && (i >
0)); i--)
{
/* Access the next format name and compare with requested */
/* format name.
*/
found_format_str = (char *) prg_list_access (pk_format_names_lptr, i - 1);
if (strcmp (found_format_str, format_str) == 0)
format_found = OPC_TRUE;
}

if (format_found == OPC_FALSE)
{
/* The requested format does not exist. Generate */
/* unformatted packets.
*/
generate_unformatted = OPC_TRUE;

/* Display an appropriate warning.
*/
op_prg_odb_print_major ("Warning from simple packet generator model
(simple_source):",
"The specified packet format",
format_str,
"is not found. Generating
unformatted packets instead.", OPC_NIL);
}

/* Destroy the lits and its elements since we don't need it */
/* anymore.
*/
prg_list_free (pk_format_names_lptr);
prg_mem_free (pk_format_names_lptr);
}

/* Make sure we have valid start and stop times, i.e. stop time is */
/* not earlier than start time.
*/
if ((stop_time <= start_time) && (stop_time != SSC_INFINITE_TIME))
{
/* Stop time is earlier than start time. Disable the source. */
start_time = SSC_INFINITE_TIME;
}

```

```

        /* Display an appropriate warning.
        */
        op_prg_odb_print_major ("Warning from simple packet generator model (simple_source):",
                                "Although the generator is not disabled
(start time is set to a finite value)",
                                "a stop time that is not later than the
start time is specified.",
                                "Disabling the generator.", OPC_NIL);
    }

    /* Schedule a self interrupt that will indicate our start time for */
    /* packet generation activities. If the source is disabled,          */
    /* schedule it at current time with the appropriate code value.      */
    if (start_time == SSC_INFINITE_TIME)
    {
        op_intrpt_schedule_self (op_sim_time (), SSC_STOP);
    }
else
    {
        op_intrpt_schedule_self (start_time, SSC_START);

        /* In this case, also schedule the interrupt when we will stop */
        /* generating packets, unless we are configured to run until    */
        /* the end of the simulation.                                     */
        /*
        if (stop_time != SSC_INFINITE_TIME)
        {
            op_intrpt_schedule_self (stop_time, SSC_STOP);
        }

        next_intarr_time = oms_dist_outcome (interarrival_dist_ptr);

        /* Make sure that interarrival time is not negative. In that case it */
        /* will be set to 0.
        */
        if (next_intarr_time < 0)
        {
            next_intarr_time = 0.0;
        }

    }

    /* Register the statistics that will be maintained by this model. */
    bits_sent_hdl = op_stat_reg ("Generator.Traffic Sent (bits/sec)",
        OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
    packets_sent_hdl = op_stat_reg ("Generator.Traffic Sent (packets/sec)",
        OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
    packet_size_hdl = op_stat_reg ("Generator.Packet Size (bits)",
        OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
    interarrivals_hdl = op_stat_reg ("Generator.Packet Interarrival Time (secs)", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

```

The Generate enter execs are shown below:

```

/* At the enter execs of the "generate" state we schedule the */
/* arrival of the next packet.
*/
if (op_sim_time () < step_time)
{
    next_intarr_time = oms_dist_outcome (interarrival_dist_ptr);
}
else
{
    next_intarr_time = oms_dist_outcome (interarrival_dist_ptr_1);
}

/* Make sure that interarrival time is not negative. In that case it */
/* will be set to 0.
*/
if (next_intarr_time < 0)
{

```

```
next_intarr_time = 0;
}

next_pk_cvh = op_intrpt_schedule_self (op_sim_time () + next_intarr_time, SSC_GENERATE);
```

A.2.2 Self-similar source

This traffic source model is constructed from 40 ON-OFF sources as shown in figure A3. The aggregator at the top right-hand side of the diagram effectively sums all of the sources.

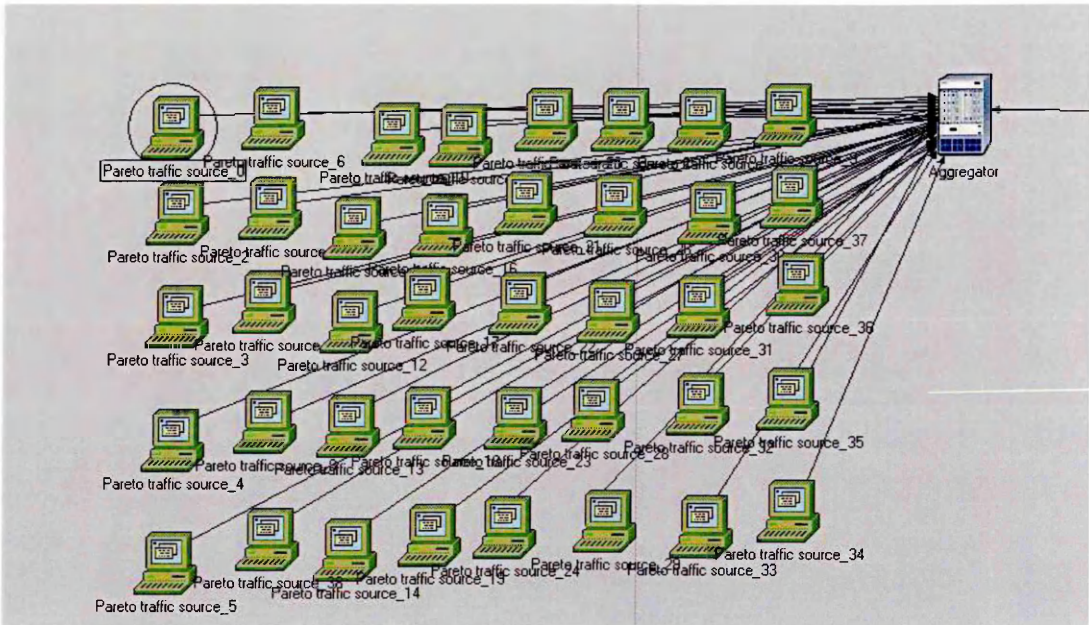


Figure A3. Compound self-similar traffic generator, network model

The node model inside each of the 40 sources is shown in figure A4

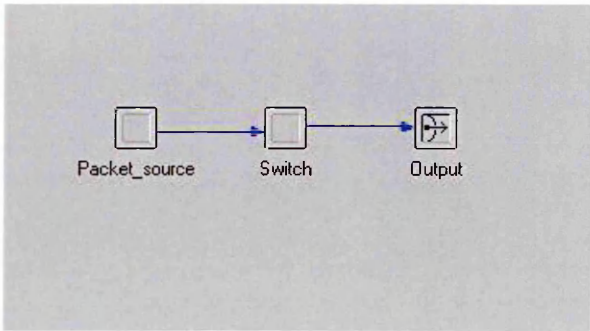


Figure A4. Node model in each source

Figure A4 shows a simple packet source in series with a switch before being output from the source. The process model within the packet source is another instance of figure A2 and accompanying modified executives so that steps in load can be generated with self-similar traffic also. The process model for the switch is shown in figure A5.

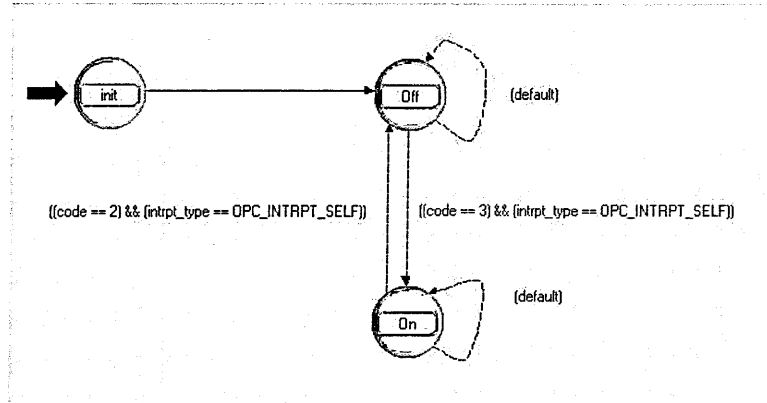


Figure A5. On-off switches in each source, process model

Like all OPNET process models, the on-off switch is a state machine. When the machine is in the top red state, the switch destroys packets and when it is in the bottom red state, the switch passes packets. The probability of transition between states is taken from independent Pareto distributions, loaded by the Init state.

This process model was developed specially for this study. Below are the executives for the states. There is no function block associated with this process.

Init state executives:

```

/* At this initial state, we read the values of source attributes. */

/* Obtain the object id of the surrounding module. */
own_id = op_id_self ();

/* Load the PDF that will be used in computing the on-off times. */

op_ima_obj_attr_get (own_id, "Position on", &position_on);
op_ima_obj_attr_get (own_id, "Shape on", &shape_on);

op_ima_obj_attr_get (own_id, "Position off", &position_off);
op_ima_obj_attr_get (own_id, "Shape off", &shape_off);

pareto_ptr_on = op_dist_load ("pareto", position_on, shape_on);
pareto_ptr_off = op_dist_load ("pareto", position_off, shape_off);

/* Register the statistics that will be maintained by this model. */
on_interval_hndl = op_stat_reg ("On interval", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
off_interval_hndl = op_stat_reg ("Off interval", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

/* At the beginning of the simulation, we need to schedule an interrupt to change */
/* state to ON, otherwise we remain permanently OFF */

off_interval = op_dist_outcome (pareto_ptr_off);
op_intrpt_schedule_self (op_sim_time () + off_interval, 3);

```

ON State executives (exit only):

```

/* In this state, we forward packets */
/* and schedule an interrupt to go to the off state */

```

```

/* Check interrupt type. If it is a packet arriving, it will be on */
/* stream 0. Acquire the packet and send it */

intrpt_type = op_intrpt_type ();
if (intrpt_type == OPC_INTRPT_STRM)
{
    pkptr = op_pk_get (0);
    op_pk_send (pkptr, 0);
}

/* If the interrupt type is self-scheduled with code = 2, we */
/* have to change to the OFF state. */
/* We schedule get a self-interrupt to come back into this ON state */

code = op_intrpt_code ();
if ((intrpt_type == OPC_INTRPT_SELF) && (code == 2))
{
    off_interval = op_dist_outcome (pareto_ptr_off);
    op_intrpt_schedule_self (op_sim_time () + off_interval, 3);
    op_stat_write (off_interval_hndl, off_interval);
}
printf ("Interrupt type is: %d\n" , intrpt_type);
printf ("Interrupt code is: %d\n" , code);

```

Off state executives (exit only)

```

/* In this state, we destroy packets */

/* Check interrupt type. If it is a packet arriving, it will be on */
/* stream 0. Acquire the packet and destroy it */

intrpt_type = op_intrpt_type ();
if (intrpt_type == OPC_INTRPT_STRM)
{
    pkptr = op_pk_get (0);
    op_pk_destroy (pkptr);
}

/* If the interrupt type is self-scheduled with code = 3, we */
/* have to change to the ON state. */
/* We schedule get a self-interrupt to come back into this OFF state */

code = op_intrpt_code ();
if ((intrpt_type == OPC_INTRPT_SELF) && (code == 3))
{
    on_interval = op_dist_outcome (pareto_ptr_on);
    op_intrpt_schedule_self (op_sim_time () + on_interval, 2);
    op_stat_write (on_interval_hndl, on_interval);
}

printf ("Interrupt type is: %d\n" , intrpt_type);
printf ("Interrupt code is: %d\n" , code);

```

Header block:

```

/* Include files. */
#include <oms_dist_support.h>

/* Macro definitions for state */
/* transitions. */
#define SWITCH_OFF (code == 2)
#define SWITCH_ON (code == 1)

```

The aggregator block in figure A3 adds up the outputs from all 40 on-off sources; in order that it can do this, it required the node model shown in figure A6.

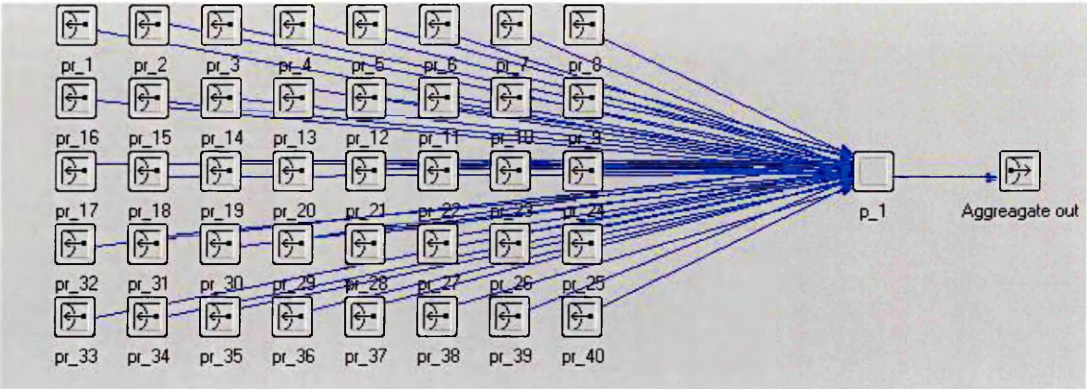


Figure A6. Aggregator node model

The process model inside p_1 was designed specifically for this study; it is very simple and shown in figure A7.

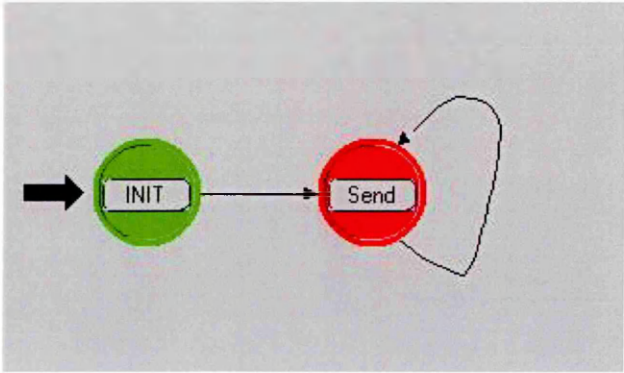


Figure A7. Aggregator process model

The executives for this process model are:

INIT:

```
/* Obtain the object identifier of the supporunding module */
my_objid = op_id_self ();

/* Register a statistic handle for the stream number */
streamhandl = op_stat_rcg ("Stream number", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
```

Send (exit only):

```
/* Get the interrupt type and check if it is a packet arriving. If it is, */
/* get the packet and the stream number (for stats purposes) */

intrpt_type = op_intrpt_type ();

if (intrpt_type == OPC_INTRPT_STRM)
{
    pkptr = op_pk_get (op_intrpt_strm ());
```

```

op_stat_write (streamhandl, op_intrpt_strm ());

/* Regardless of the input stream, send the packet out on stream 0 */

op_pk_send (pkptr, 0);
}

```

The burst switches are present in figure A1 in order that bursts of a specified number of packets can be sent. These were incorporated early in the model development to enable the user and trace packets to be switched off to check the operation of the sub-queues in the gateway. They have a fairly simple process model that was designed specifically for this study, shown in figure A8.

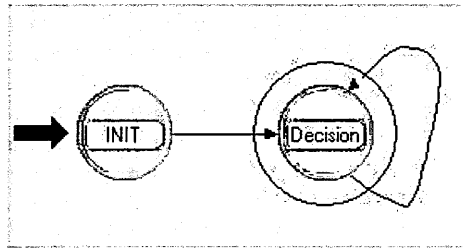


Figure A8 Burst process model

The executives are:

INIT:

```

/* Obtain the object identifier of the supporunding module */
my_objid = op_id_self ();

```

```

/* Obtain the simulation attribute for the requested burst size (in */
/* number of packets) */

```

```

op_ima_sim_attr_get (OPC_IMA_INTEGER, "User Packet burst size", &user_burst_size);

```

Decision (exit only):

```

/* Get the interrupt type and check if it is a packet arriving. If it is, it will be */
/* on stream 0. Acquire the packet. */

```

```

intrpt_type = op_intrpt_type ();

```

```

if (intrpt_type == OPC_INTRPT_STRM)
{
    pkptr = op_pk_get (0);
}

```

```

/* We need to determine whether to foward the packet or to destroy it, */
/* depending on the requested burst size */

```

```

op_pk_format_info_get (pkptr, OPC_PK_PROPERTY_FORMAT_NAME, &format_name);

```

```

if (strcmp (format_name, "mrf_user") == 0)
{
    total = total + 1;
}

```

```

if (total >= user_burst_size)
{
    op_pk_destroy (pkptr);
}

```

```
else
{
    op_pk_send (pkptr, 0);
}
```

A.3 Gateway model

The gateway node model is the most complicated in figure A1. Its node model is shown in figure A9.

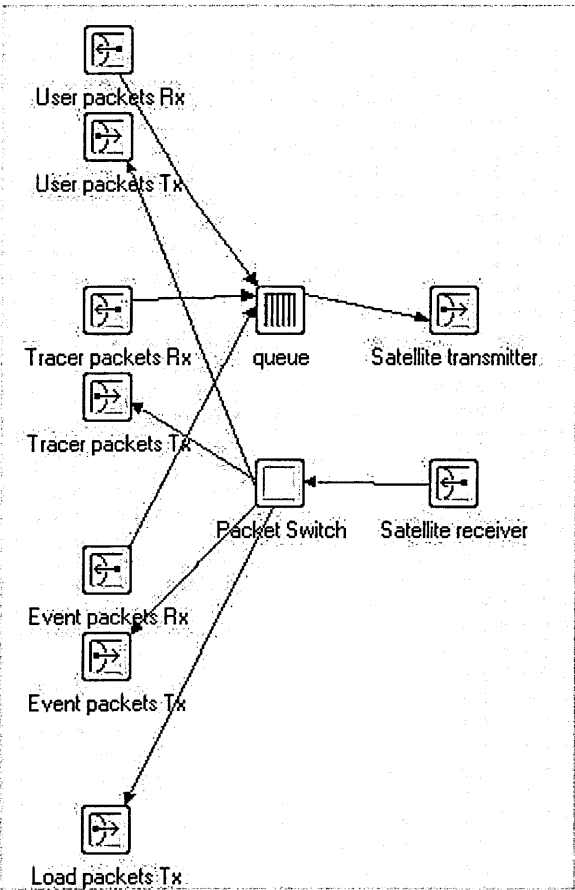


Figure A9. Gateway and Remote Network node model

The gateway node model is two-way and is identical to the Remote network node model. These models were made two-way in case it was thought necessary to model a two-way satellite link, this was not implemented.

The queue process model from the node model in figure A9 is shown in figure A10.

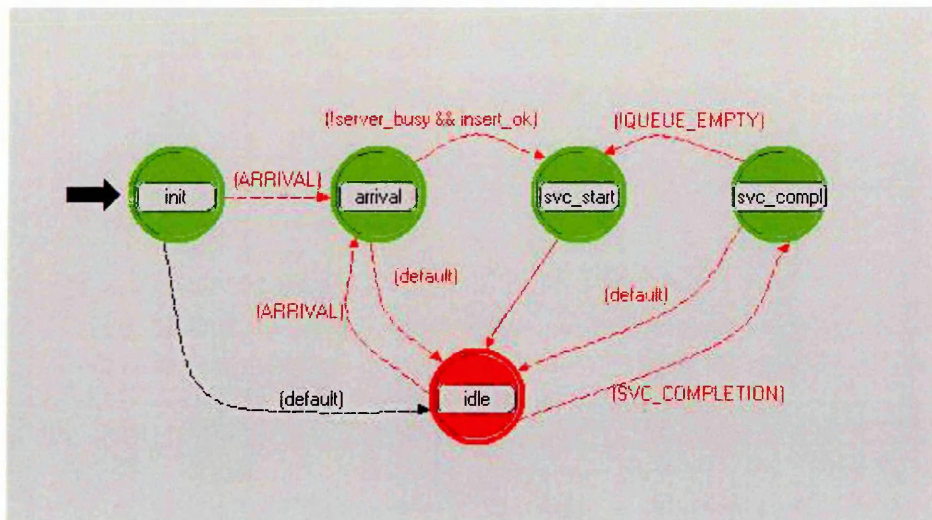


Figure A10. Queue process model in the gateway and remote network

The process model in figure A10 is based on an OPNET FIFO queue process from the OPNET v9 library, however it has been substantially modified. It now contains sub-queues with priority queueing, so that event management traffic is stored in a high priority sub-queue, with user and trace packets being stored in a lower priority sub-queue. It is also possible to set different service rates for user traffic and management traffic, along with maximum sub-queue size, from attributes set at simulation run-time and read using the Init state. It is this mechanism that was used to simulate finite queue limits to more effectively model a practical network.

The executives are shown below for this process model:

Init:

```

/* initially the server is idle          */
server_busy = 0;

/* get queue module's own object id     */
own_id = op_id_self ();

/* get simulation attributes             */
op_ima_sim_attr_get (OPC_IMA_DOUBLE, "Service rate user gateway", &user_service_rate);
op_ima_sim_attr_get (OPC_IMA_DOUBLE, "Service rate management gateway", &man_service_rate);
op_ima_sim_attr_get (OPC_IMA_INTEGER, "Subqueue max size", &max_num_bits);

/* register statistic handles */

stream_num_handl = op_stat_reg ("Stream number", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
subq_size_handl = op_stat_reg ("Subq size", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
subql_size_handl = op_stat_reg ("Subql size", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
inter_arrival_time_handl = op_stat_reg ("Inter_arrival_times", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

```

Arrival:

```

/* The transition condition into this state ensures that */
/* the interrupt is an arriving packet. */
/*Get the stream number and acquire the packet */
stream_number = op_intrpt_strm ();
pkptr = op_pk_get (stream_number);

/* Collect stream number as a statistic */
op_stat_write (stream_num_handl, stream_number);

```

```

/* Calculate the inter-arrival time between the current packet and the last packet */
/* This is a test tool for checking distributions */

pk_create_time_last = pk_create_time;
pk_create_time = op_pk_creation_time_get (pkptr);
inter_arrival_time = pk_create_time - pk_create_time_last;

/* Collect inter-arrival times as a statistic */

op_stat_write (inter_arrival_time_handl, inter_arrival_time);

/* Overhead management packets get priority. These packets arrive on stream 2 */
/* The user packets and user-accompanied packets arrive on streams 0 and 1 and have */
/* the same priority. Stream 2 packets are put into subq0 and the rest into subq1. */

if (stream_number == 2)
{
    subq_index = 0;
}
else
{
    subq_index = 1;
}

/* Check size of subqueues. If already full, set a full flag */
num_bits = op_subq_stat (subq_index, OPC_QSTAT_BITSIZE);
if (num_bits >= max_num_bits)
{
    full_flag = 1;
}
else
{
    full_flag = 0;
}

/* attempt to place the packet in the appropriate subqueue */

if (full_flag == 1)
{
    /* do not insert packet into sub-queue. Set flag indicating */
    /* insertion fail and deallocate the packet */
    insert_ok = 0;
    op_pk_destroy (pkptr);
}
else
{
    if (op_subq_pk_insert (subq_index, pkptr, OPC_QPOS_TAIL) != OPC_QINS_OK)
    {
        /* the inserton failed (due to to a */
        /* full queue or some other reason) deallocate the packet. */
        op_pk_destroy (pkptr);

        /* set flag indicating insertion fail */
        /* this flag is used to determine */
        /* transition out of this state */
        insert_ok = 0;
    }
    else
    {
        /* insertion was successful */
        insert_ok = 1;
    }
}

Svc_start:
/* subqueue 0 gets priority. */

if (op_subq_empty(0) == OPC_FALSE)
{

```

```

/* get a handle on packet at head of subqueue 0 */
/* (this does not remove the packet) */
pkptr = op_subq_pk_access (0, OPC_QPOS_HEAD);
op_stat_write (subq_size_handl, OPC_QSTAT_PKSIZE);

/* determine the packets length (in bits) */
pk_len = op_pk_total_size_get (pkptr);

/* determine the time required to complete */
/* service of this management packet */
pk_svc_time = pk_len / man_service_rate;

/* schedule an interrupt for this process */
/* at the time where service ends. */
op_intrpt_schedule_self (op_sim_time () + pk_svc_time, 0);
}
else
{
/* get a handle on packet at head of subqueue 1 */
pkptr = op_subq_pk_access (1, OPC_QPOS_HEAD);
op_stat_write (subq1_size_handl, OPC_QSTAT_PKSIZE);

/* determine the packets length (in bits) */
pk_len = op_pk_total_size_get (pkptr);

/* determine the time required to complete */
/* service of this user packet */
pk_svc_time = pk_len / user_service_rate;

/* schedule an interrupt for this process */
/* at the time where service ends. */
op_intrpt_schedule_self (op_sim_time () + pk_svc_time, 1);
}

/* the server is now busy. */
server_busy = 1;

Svc_compl:
/* extract packet at head of queue; this */
/* is the packet just finishing service */
/* We have to choose the right subqueue. The interrupt */
/* code is set to 0 if the packet finishing service is in */
/* subqueue 0 */

if (op_intrpt_code () == 0)
{
pkptr = op_subq_pk_remove (0, OPC_QPOS_HEAD);
}
else
{
pkptr = op_subq_pk_remove (1, OPC_QPOS_HEAD);
delay = op_subq_stat (1, OPC_QSTAT_DELAY);
}

/* forward the packet on stream 0 */
op_pk_send_forced (pkptr, 0);

/* server is idle again. */
server_busy = 0;

```


A.4 Packet switch model

The packet switch process model from figure A9 is shown in figure A11.

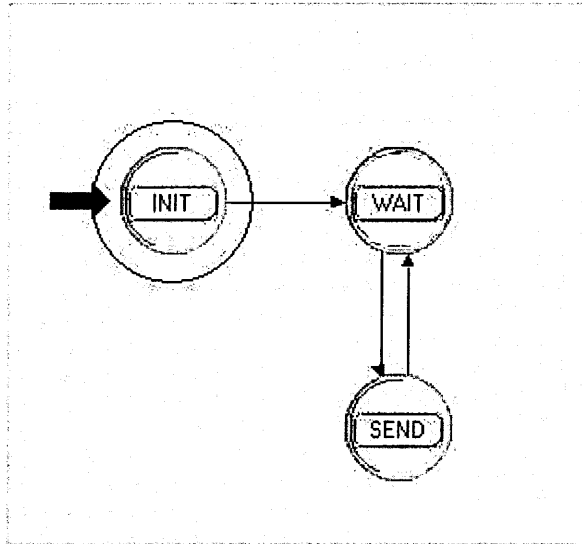


Figure A11. Packet switch process model

The packet switch process model is quite simple, but the code within it has to decide which stream to send the packet, depending on packet type. If it is a user packet the model directs it towards the user_client on stream 0, if a trace packet (mrf_oss_pkt) the model directs it towards the trace_client on stream 1. It also has the provision to detect a third type of packet called 'load' to enable the model to be loaded with background data at a later date and a fourth type for any particular applications that might be running.

The INIT state has just one purpose, to register a statistic handle for recording the stream on which the packet is sent, for diagnostic purposes.

INIT :

```
/* Obtain the object identifier of the supporunding module */
my_objid = op_id_self ();

/* Register the statistics handles that will be used for packet statistics */
stream_index_handl = op_stat_reg("Stream Index", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
```

Wait:

The process model waits in this state between interrupts, the following c-code is in the exit executives from this state, and makes the decision on the correct output stream:

```
/* Get the interrupt type and check if it is a packet arriving. If it is, it will be */
/* on stream 0. Acquire the packet. */

intrpt_type = op_intrpt_type ();

if (intrpt_type == OPC_INTRPT_STRM)
{
    pkptr = op_pk_get (0);
```

```

    }

    /* We need to determine whether the packet is a user packet, a tracer packet, or a packet */
    /* from an FTP (alarm event) session (high priority) */

    op_pk_format_info_get (pkptr, OPC_PK_PROPERTY_FORMAT_NAME, &format_name);

    if (strcmp (format_name, "mrf_user") == 0)
    /* the packet is a user packet (format mrf_user) - (strcmp returns false) */
    {
        out_stream = 0;
    }
    else
    {
        if (strcmp (format_name, "mrf_oss_pckt") == 0)
        {
            /* the packet is a tracer packet */
            out_stream = 1;
        }
        else
        {
            if (strcmp (format_name, "mrf_load_pckt") == 0)
            {
                /* the packet is a loading packet */
                out_stream = 3;
            }
            else
            {
                /* the packet is another type (eg an application) */
                out_stream = 2;
            }
        }
    }

    op_stat_write (stream_index_handl, out_stream);

Send:
    /* In this state, all packets are simply forwarded on          */
    /* to the next link within the appropriate stream */

    op_pk_send (pkptr, out_stream);

```

The Send state could equally well be replaced by a transition action to send the packet.

The OPNET model is also used with a set of probes to monitor throughput on links and within node models, constructed using the probe editor.

End of Appendix A

Appendix B: Basic queuing theory

B.1 Queueing theory

A simple queue and server is shown in figure B1.

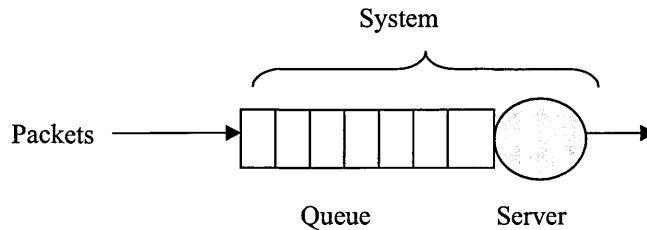


Figure B1. Simple queue and server

The mean arrival rate of packets into the queue is given the symbol λ . The mean rate at which packets are serviced by the server, i.e. taken from the head of the queue, is given the symbol μ . The system is stable only when $\lambda < \mu$. The 'system' is the combination of queue and server.

The mathematics in this appendix is based on literature [B1, B2], but additional detail and steps have been added, since it was not always clear how certain results were arrived at. For example, further down the text a recurrence relation is used followed by a manipulation involving a Maclaurin's series which are additional steps to get to the results quoted in the literature.

The next step is to consider a birth-death process where the birth is the arrival of a packet and the death is departure of a packet from the system. A time interval Δt is defined where the probability of a birth or death happening is $\lambda \Delta t$ and $\mu \Delta t$ respectively, with the following conditions:

- the probability of more than one birth or death in the interval Δt is vanishingly small
- there cannot be a death unless the population size is at least one.

Since the probability that there is more than one birth or death in the interval Δt is set to zero, the probability of zero births or deaths in the interval is $(1 - \lambda \Delta t)$ and $(1 - \mu \Delta t)$ respectively.

Let the number of packets in the system be N , so that $N = 0$ is an empty system.

Let the probability of there being N packets in the system at time t be denoted $P_N(t)$.

If time moves on by one interval Δt , the number of packets in the system at this time is $P_N(t + \Delta t)$. The probability of $N = 0$ at $(t + \Delta t)$ is given by the probability that [$N = 0$ at time t AND zero births during the interval] OR [$N = 1$ at time t and one death during the interval], i.e.

$$P_0(t + \Delta t) = P_0(t)[1 - \lambda \Delta t] + P_1(t) \mu \Delta t$$

The probability of N being any particular value (> 0) at $(t + \Delta t)$ is:

- N being the same as at t with no births AND no deaths
- N being the same as at t with one birth AND one death
- N decreasing from that at t with no births AND one death
- N increasing from that at t with one birth AND no deaths

Writing this out, the four terms are

$$P_N(t + \Delta t) = P_N(t)(1 - \lambda\Delta t)(1 - \mu\Delta t) + (\lambda\mu\Delta t^2) + P_{N-1}(t)\lambda\Delta t + P_{N+1}(t)\mu\Delta t$$

Multiplying out the brackets and ignoring Δt^2 terms,

$$P_N(t + \Delta t) = P_N(t)(1 - \lambda\Delta t - \mu\Delta t) + P_{N-1}(t)\lambda\Delta t + P_{N+1}(t)\mu\Delta t$$

Letting $\Delta t \rightarrow 0$, for the two cases of $N = 0$ and $N \neq 0$,

$$\frac{dP_0(t)}{dt} = -\lambda P_0(t) + \mu P_1(t) \quad (\text{B-1})$$

and

$$\frac{dP_N(t)}{dt} = -(\lambda + \mu)P_N(t) + \lambda P_{N-1}(t) + \mu P_{N+1}(t) \quad (\text{B-2})$$

These differential equations have transient and steady state solutions. The literature assumes equilibrium and hence the LHS of the above equations are set to zero.

Define the ratio $\frac{\lambda}{\mu}$ as the traffic intensity measured in Erlangs with the symbol ρ , then for stability, $\rho \leq 1$. This is also referred to as the queue ‘utilisation’ or ‘load’ in the literature.

From equation (B-1), $\mu P_1(t) = \lambda P_0(t)$, so that $P_1 = \rho P_0$

From equation (B-2), $\mu P_{N+1}(t) = (\lambda + \mu)P_N(t) - \lambda P_{N-1}(t)$

Divide equation (B-2) by μ gives $P_{N+1}(t) = (1 + \rho)P_N(t) - \rho P_{N-1}(t)$

This is a recurrence relation whose solution is

$$P_i = \rho^i (1 - \rho)$$

with the normalisation condition $\sum P_i = 1$. This holds for $\rho \leq 1$.

The mean number of packets in the system is

$$N = \sum_{i=0}^{\infty} iP_i = \sum_{i=0}^{\infty} i\rho^i(1-\rho)$$

If this sum is evaluated for $i = 0, 1, 2, 3$ etc the first few terms are

$$\begin{aligned} N &= 0 + \rho(1-\rho) + 2\rho^2(1-\rho) + 3\rho^3(1-\rho) + \dots \\ &= 0 + \rho - \rho^2 + 2\rho^2 - 2\rho^3 + 3\rho^3 - 3\rho^4 + 4\rho^4 + \dots \end{aligned}$$

Gathering powers of ρ gives

$$\begin{aligned} N &= \rho + \rho^2 + \rho^3 + \rho^4 + \dots \\ &= \rho(1 + \rho + \rho^2 + \rho^3 + \dots) \end{aligned}$$

But this is a Maclaurin's series which, for $\rho < 1$, is equal to $N = \frac{\rho}{1-\rho}$, which is the result given in the literature [B1].

B.2 Derivation of the Poisson arrival process

Proceeding much as before (but considering just the arrivals), the probability that there are N arrivals in the system at time $(t + \Delta t)$ is either that there have been no arrivals since time t OR that there has been one arrival in the interval Δt from when there were $N-1$ occupants. Writing this out, we get

$$P_N(t + \Delta t) = P_N(t)[1 - \lambda\Delta t] + P_{N-1}(t)\lambda\Delta t \text{ for } N = 1, 2, \dots$$

and with $\Delta t \rightarrow 0$ we get

$$\frac{dP_0(t)}{dt} = -\lambda t \quad (\text{B-3})$$

and

$$\frac{dP_N(t)}{dt} = -\lambda[P_N(t) - P_{N-1}(t)] \quad (\text{B-4})$$

In fact equations (B-3) and (B-4) are the same as (B-1) and (B-2) but with the service terms missing.

The solution to equation (B-3) is, by inspection,

$$P_0(t) = e^{-\lambda t} \quad (\text{B-5})$$

If we insert this into equation (B-4) for $N = 1$, we get

$$\frac{dP_1(t)}{dt} = -\lambda P_1(t) + \lambda e^{-\lambda t}$$

and the solution to this differential equation is

$$P_1(t) = \lambda t e^{-\lambda t}$$

Continuing by induction, the solution to equation (B-4) is

$$P_N(t) = \frac{(\lambda t)^N e^{-\lambda t}}{N!} \quad (\text{B-6})$$

Equation (B-6) is the Poisson distribution. It is an arrival or 'birth' process, with mean rate λ and can be used to model a wide range of natural physical and organic processes. It is also memoryless, in that the probability of an arrival or birth during a time interval is not affected by previous arrivals.

B.3 Derivation of time spent in the system

The discussion up to now has been about the system state and the arrival process. What is of more interest to this study however is the time spent in the system since this impacts on the Quality of Service. The Pollackzek-Khinchin (P-K) equation for mean time spent in the system is now derived.

A famous result by Little, which holds for virtually all queueing systems [B1] is

$$N = \lambda W$$

where N and W are the number of packets and waiting time in the system respectively. It is quite intuitive, i.e. the mean number in the system is the mean arrival rate multiplied by the mean waiting time. But we need the distribution of the waiting time and, to begin, let τ be the random variable between adjacent packet arrivals, whose probability distribution function (PDF) is $A(t)$ and the probability density function (pdf) is $a(t)$.

$A(t)$ is the probability that the time between arrivals is $\leq \tau$, i.e.,

$$A(t) = 1 - P(\tau > t)$$

But $P(\tau > t)$ is the probability that no arrivals occur in the interval $(0, t)$, so that

$$A(t) = 1 - P_0(t)$$

Which, by reference to equation (B-5) is

$$A(t) = 1 - e^{-\lambda t}$$

Differentiating this to obtain the pdf:

$$a(t) = \lambda e^{-\lambda t}$$

The next step in obtaining the P-K formula is to establish the PASTA property (Poisson Arrivals See Time Average). The way this works is that packets arriving at a non-empty queue will see:

- one or more packets in the queue,
- a packet currently being transmitted from the output of the queue (being serviced), which will have a residual time left.

The queue is assumed to be First In First Out (FIFO), which means that packets are not re-ordered in the queue.

The residual service times from the PASTA property is now used to develop a method of analysis for the M/G/1 queue, where the mean of the residual time is R . Therefore, the P-K equation is only valid for this type of queue; it holds for Markov (memoryless) arrival patterns (such as Poisson) but not for arrival patterns with any degree of self-similarity.

Bearing in mind these restrictions and continuing, the mean waiting time in the queue is the number of packets waiting multiplied by the mean service time, plus a residual service time, i.e.

$$\overline{W}_q = N_q E\{X\} + R$$

where $E\{X\}$ is the expected service time ($= 1/\mu$) and \overline{W}_q is the mean waiting time in the queue.

Using Little's result $N = \lambda W$,

$$\overline{W}_q = \lambda \overline{W}_q E\{X\} + R$$

Putting $\rho = \lambda/\mu = \lambda E\{X\}$, and dividing by \overline{W}_q gives

$$\overline{W}_q = \frac{R}{1-\rho} \quad (\text{B-7})$$

The residual service time R can be found using a graphical approach. In figure B2 below, the y-axis $r(\tau)$ is the value of the residual transmission time when the system is examined at time τ . The sudden vertical jumps in the graph happen when a packet departs the queue and begins transmission (or service), which is then at a constant rate. They are different heights if the packets are different lengths. We can arbitrarily make the service rate a slope of -1 on the graph, so that the vertical and horizontal sides are equal in length.

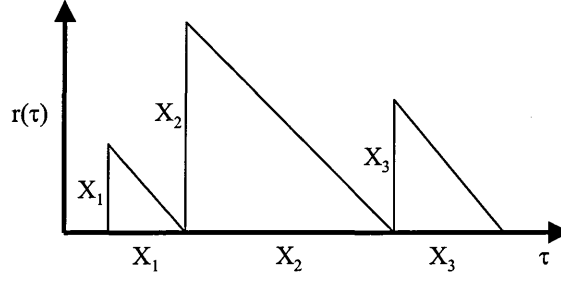


Figure B2. Graphical representation of residual service time

Let $M(t)$ be the number of departures from the queue in time interval $(0, t)$

The time average of $r(\tau)$ in the interval $(0, t)$ is

$$\frac{1}{t} \int_0^t r(\tau) d\tau \approx \frac{1}{t} \sum_{i=1}^{M(t)} \frac{1}{2} X_i^2 \quad (= \text{area of triangles})$$

The approximation is due to the last residual time in $(0, t)$ being ignored; the approximation becomes exact as $t \rightarrow \infty$.

Re-arranging and performing a small trick of multiplying by $M(t)/M(t)$, the time average of $r(\tau)$ becomes

$$r(\tau) = \frac{1}{2} \frac{M(t)}{t} \frac{1}{M(t)} \sum_{i=1}^{M(t)} X_i^2$$

Actually a mistake by Bose has been corrected here in this equation, he included a spurious factor of $\frac{1}{2}$.

For $t \rightarrow \infty$, the time average of $r(\tau) = R$, $\frac{M(t)}{t} \rightarrow \lambda$ and $\frac{1}{M(t)} \sum_{i=1}^{M(t)} X_i^2 \rightarrow X^2$

Therefore by PASTA, $R = \frac{1}{2} \lambda X^2$

Combining this with equation (B-7),

$$\overline{W}_q = \frac{\lambda X^2}{2(1-\rho)} \quad (\text{B-8})$$

Equation (B-8) is the wanted P-K mean value formula and gives a value for the waiting time in the queue. It is derived using the Takács recurrence relation in the main text.

To obtain the mean delay through the system, the mean service time must be added. Also, the mean service time X can be replaced by $1/\mu$, to give perhaps a more useful form of the equation for the system delay W_s

$$\overline{W_s} = \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)} .$$

B.4 References

[B1] Sanjay Bose, *An Introduction to Queueing Systems*. Kluwer Academic / Plenum Publishers, New York, 2002.

[B2] Leonard Kleinrock, *Queueing Systems, Volume 1: Theory*. John Wiley & Sons, 1975.

End of Appendix B

Appendix C Test network router configurations and RUDE files

C.1 Test network

The test network configuration is shown in figure C-1, with IP addresses.

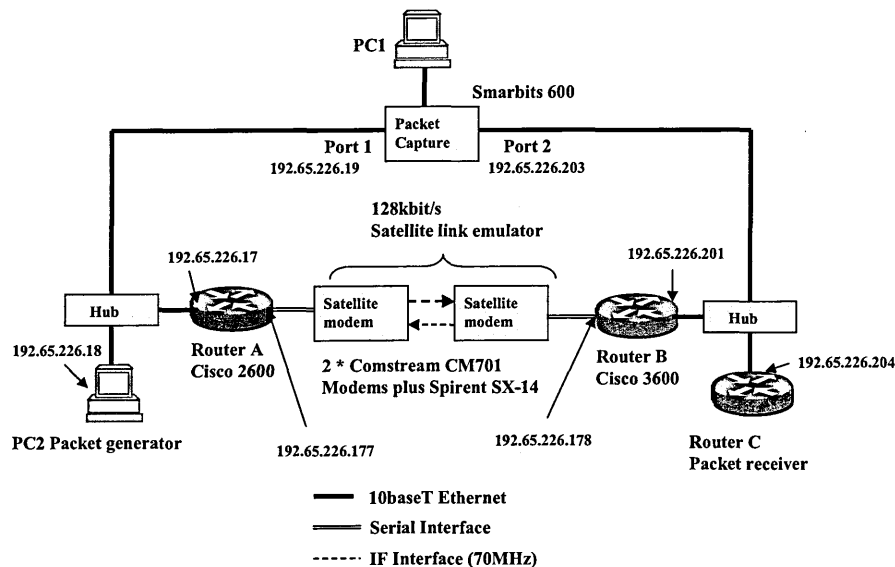


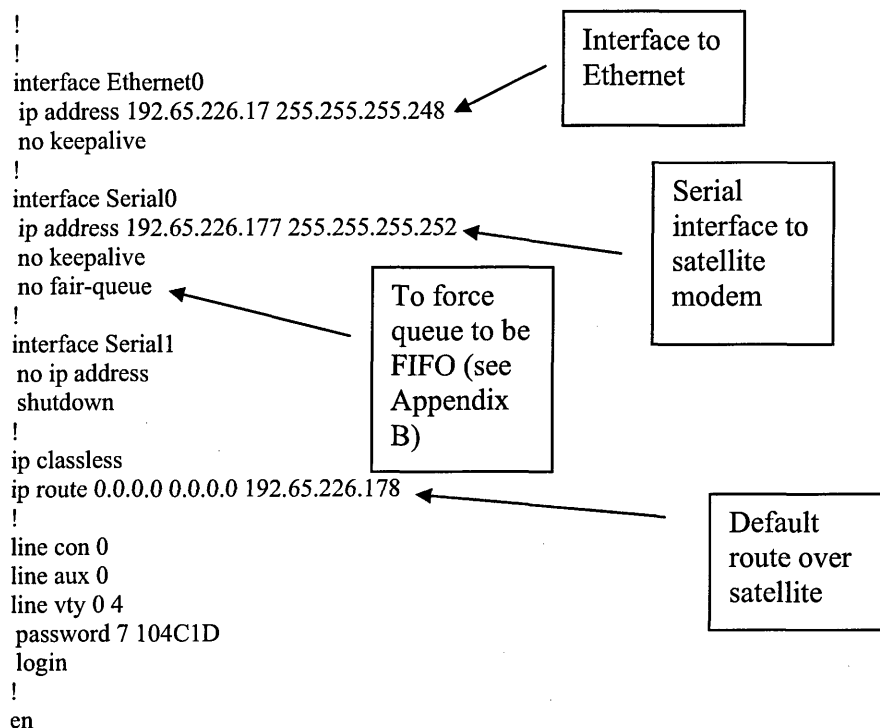
Figure C1 Test network diagram

C.2 Router Configurations

Router A is shown, router B is very similar. There are three subnets created, one on each network at either end of figure C1, and the other is between the routers over the satellite link / emulator.

Current configuration:

```
!  
version 11.2  
service password-encryption  
no service udp-small-servers  
no service tcp-small-servers  
!  
hostname TOP  
!  
enable secret 5 $1SqTH0$T9IR0B9gpxroeTr6.gSQe0  
enable password 7 020410  
!  
ip subnet-zero
```



C.3 RUDE Files

The script used to run RUDE is constructed as follows

```

START NOW
1000 0001 ON 3001 192.65.226.204:10001 TRACE Exp_5.txt
1500 0002 ON 3002 192.65.226.204:10002 TRACE 44_h.txt
139500 0002 OFF
140000 0001 OFF

```

The first line tells RUDE to start generating packets immediately. The 'NOW' can be replaced by an absolute time if required.

The second line tells RUDE to start generating flow 0001 at 1000ms after the START with the ON command. The '3001' is the sending port number. The destination IP address and port number is 192.65.226.204:10001. The TRACE options tells RUDE to read down a file to produce packets. The TRACE files that are called by the script enable RUDE to generate packets of any length with any inter-arrival interval.

The last 2 lines of the script switch OFF the 0002 and 0001 flows at 139.5 seconds and 140 seconds respectively.

A fragment of the TRACE file called 'Exp_5.txt' that is called by the script above and generates Poisson distributed traffic with $\lambda = 0.5$ and a packet length of 556 bytes (not counting headers) is as follows:

556	0.111377
556	0.049202
556	0.008689
556	0.012158
556	0.024928
556	0.019444
556	0.114475
556	0.041513
556	0.083063
556	0.013410
556	0.041122
556	0.148161
556	0.015601
556	0.065505
556	0.025602
556	0.032563
556	0.090613

Column 1 gives the packet length while column 2 gives the inter-arrival times in seconds. It was found that this was accurate to within $2\mu\text{s}$ as measured using the Smartbits packet capture. The actual TRACE files used were several thousand lines, one for each packet, generated using OPNET. Similar files were used to generate self-similar flows.

The trace packets were generated using another TRACE file in line 3 of the script, a fragment of the '44_h.txt' file that generates 64 byte trace packets (44 byte plus 20 byte header) at a constant rate of one every 5.14 seconds is as follows:

44	5.14
44	5.14
44	5.14
44	5.14
44	5.14
44	5.14
44	5.14
44	5.14
44	5.14

End of Appendix C